

UNIVERSITÉ DU QUÉBEC À TROIS RIVIÈRES

MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC
À TROIS RIVIÈRES

COMME EXIGENCE PARTIELLE DE LA MAÎTRISE
EN ÉLECTRONIQUE INDUSTRIELLE

PAR
ARNAUD LUCAS DE COUVILLE

MESURE ET FUSION DE DONNÉES
ULTRASONORES POUR LA PERCEPTION D'OBJET
APPLIQUÉES AUX SYSTÈMES MOBILES

SEPTEMBRE 1999

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Résumé

Les systèmes mobiles : véhicules autonomes, robots mobiles et autres permettent de remplacer l'humain dans des tâches dangereuses ou répétitives. Un des grands défis de la science est de faire évoluer ces dispositifs dans des milieux où l'homme est présent et où ils peuvent être des sources de collision. La perception est la discipline qui détecte les éléments de l'environnement.

Ce projet a pour but de réaliser un système de localisation des objets dans l'environnement proche d'un système mobile. Ce système utilise les ultrasons comme moyen de détection. Une étude sur la propagation ultrasonore et les capteurs ultrasonores permet de cerner les avantages et les inconvénients de cette source sensorielle. Le placement des objets s'effectue en deux étapes: la première consiste à mesurer la distance qui sépare l'objet du capteur; la seconde permet à partir de plusieurs mesures de la première de déterminer précisément sa position à l'aide d'un algorithme de fusion.

Ce mémoire comporte les travaux suivants : le développement de deux algorithmes pour la mesure de distance et la fusion de ces distances pour positionner précisément les objets détectés; la réalisation d'un module de distance effectué à l'aide d'un micro-contrôleur 16 bits du constructeur Intel; une proposition d'un système multi-capteurs.

Les résultats obtenus sont robustes aux bruits, fiables et précis. Une attention particulière est portée pour rendre le système peu complexe aussi bien au niveau matériel que logiciel.

Remerciements

Je tiens à remercier mon directeur de recherche, professeur Pierre Sicard, pour son expertise, sa disponibilité et ses encouragements qui m'ont aidé à réaliser ce mémoire.

Je tiens aussi à remercier le professeur Yves Dubé qui a accepté de co-diriger ma recherche et m'a proposé d'appliquer mes connaissances dans un projet industriel.

Je remercie également, tous les professeurs qui ont contribué à ma formation avec une mention particulière pour les conseils pratiques du professeur Kodjo Agbossou, au soutien de ma famille et de mes amis.

Table des matières

Résumé	i
Remerciements	ii
Table des matières.....	iii
Liste des figures.....	ix
Liste des tableaux	xii
Liste des symboles et abréviations.....	xiii
Chapitre 1 : Introduction	1
1.1 CONTEXTE	1
1.2 PROBLÉMATIQUE	3
1.3 OBJECTIFS.....	4
1.4 LES MÉTHODES EXISTANTES	5
1.5 MÉTHODOLOGIE	8
1.6 ORGANISATION DU MÉMOIRE.....	8
Chapitre 2 : Système de mesure ultrasonore	10
2.1 GÉNÉRALITÉS	10
2.2 PROPAGATION ULTRASONORE.....	13
2.2.1 La réflexion.....	13
2.2.2 Les réflexions et leur problématique	15
2.2.3 Les propriétés acoustiques	16
2.3 MODÉLISATION DU TRANSDUCTEUR	21
2.3.1 Le transducteur	21
2.3.2 Choisir son transducteur	21

2.3.3 Modèle physique.....	24
2.3.4 Modèle électrique	26
2.4 TECHNIQUES DE MESURE DE LA DISTANCE	27
2.4.1 Mesure par le temps de vol	27
2.4.2 Mesure par la différence de phase	28
2.5 COMPOSITION D'UN SYSTÈME DE MESURE ULTRASONORE	30
2.5.1 Principe de fonctionnement du système Polaroid.....	30
2.5.2 Décomposition sous forme d'un système de mesure.....	30
2.5.3 Décomposition en blocs élémentaires d'un système de mesure	31
Chapitre 3 : Algorithme de reconstitution de la distance et de fusion	35
3.1 RECONSTITUTION DE LA DISTANCE	35
3.1.1 Principe	35
3.1.2 Simulation.....	38
3.1.3 Algorithme de mesure de distance.....	43
3.2 ALGORITHME DE FUSION.....	45
3.2.1 Principe	45
3.2.2 Algorithme de Kalman étendu pour la perception d'objet	48
3.2.3 Simulation.....	55
Chapitre 4 :Schéma et implantation	72
4.1 ÉTUDE DU BLOC DE CONVERSION.....	72
4.1.1 Principe	72
4.1.2 Analyse fonctionnelle	72
4.1.3 Analyse structurelle	73
4.2 ÉTUDE DU BLOC DE RECONSTITUTION	78
4.2.1 Principe	78
4.2.2 Cahier des charges pour une structure matérielle	78
4.2.3 Choix de la structure matérielle	79
4.2.4 Justification des modes du 8XC196KD pour l'application.....	87
4.3 IMPLANTATION DE L'ALGORITHME	90

4.3.1 Algorithme du programme principal	90
4.3.2 Algorithme de la routine émission déclenchée par HSI.0 externe (INT04)	90
4.3.3 Algorithme de la routine fin de conversion	91
4.4 STRUCTURE ENVISAGÉE POUR L'ALGORITHME DE RECONSTITUTION DE POSITION	92
Chapitre 5 : Résultats expérimentaux	93
5.1 MÉTHODOLOGIE	93
5.2 RÉSULTATS DES TESTS INDIVIDUELS	94
5.3 TESTS GÉNÉRAUX	95
5.3.1 Détermination du temps d'acquisition le plus long.....	95
5.3.2 Détermination de la résolution.....	95
5.3.3 Détermination de la fidélité du système	95
5.3.4 Détermination de la célérité	96
5.3.5 Calibration	96
5.3.6 Les signaux électriques.....	99
5.3.7 Action de l'alignement et des très faibles mesures	102
5.4 CONCLUSION	104
Chapitre 6 : Proposition d'un système multi-capteurs	105
6.1 DÉFINITION DE L'APPLICATION.....	105
6.2 DÉFINITION DE LA STRUCTURE.....	106
6.3 AGENCEMENT DES TRANSDUCTEURS.	109
6.4 DISCUSSION SUR LE MOYEN DE COMMUNICATION.....	110
6.4.1 Type de liaison.....	110
6.4.2 Les interfaces séries	111
6.4.3 Les interfaces parallèles.....	113
6.4.4 Le protocole	114
6.5 DÉFINITION DE LA STRUCTURE PROPOSÉE	116
Chapitre 7 : Conclusion.....	122
7.1 BILAN	122
7.2 CONTRIBUTIONS	123

7.3 APPLICATIONS ET DÉVELOPPEMENTS FUTURS	123
Bibliographie	125
 Annexe A : Compléments ultrasonores	A-1
A.1. MODELISATION DU TRANSDUCTEUR (COMPLEMENTS CHAPITRE2)	A-1
A.1.1. Le modèle électrique en transmission.....	A-1
A.1.2. Le modèle électrique en réception	A-3
A.1.3. Le modèle électrique d'un transducteur qui détecte un objet.....	A-4
A.2. CALCUL DU COEFFICIENT D'ABSORPTION DE L'AIR	A-5
A.3. APPLICATION NUMERIQUE DE L'ATTENUATION TOTALE (POUR PLUSIEURS DISTANCES.....)	A-6
A.4. COMMENT CHOISIR SON TRANSDUCTEUR ?	A-7
A.4.1. Principe	A-7
A.4.2. Les sensibilités	A-8
A.4.3. Les pertes en vol	A-8
A.4.4. La sensibilité totale	A-9
 Annexe B : Compléments pour l'algorithme de mesure de distance	B-1
B.1. CHOIX DES FONCTIONS DE PRETRAITEMENT	B-1
B.2. INFLUENCE DU NOMBRE DE CYCLES EMIS	B-2
B.3. INFLUENCE DE LA FREQUENCE D'ACQUISITION.....	B-5
B.4. INFLUENCE DU SECOND ECHO.....	B-6
 Annexe C : Compléments pour l'algorithme d'estimation de la tangente ...	C-1
.....	C-1
C.1. CHANGEMENT DE REFERENTIEL	C-1
C.2. SIMULATEUR POUR LA FUSION DE DONNEES	C-2
C.2.1. Création du simulateur	C-2
C.2.2. Générateur de trajectoire	C-3
C.2.3. Générateur des positions des deux transducteurs.....	C-5

C.2.4. Générateur de données ultrasonores.....	C-5
C.2.5. Initialisation du simulateur.....	C-7
C.2.6. Vérification du choix des facteurs d'oubli.....	C-7
C.2.7. Estimation des paramètres.....	C-8
C.2.8. Choix des algorithmes.....	C-9
C.3. LES PROGRAMMES MATLAB.....	C-10
C.3.1. Programme principal : simulateur.m.....	C-10
C.3.2. Sous-programme : genetraj.m.....	C-14
C.3.3. Sous-programme : genepostr.m.....	C-15
C.3.4. Sous-programme : genedonus.m.....	C-16
C.3.5. Sous-programme : inisimu.m.....	C-17
C.3.6. Sous-programme : inikal.m.....	C-20
C.3.7. Sous-programme : factverif.m.....	C-21
C.3.8. Sous-programme : modgeo.m.....	C-22
C.3.9. Sous-programme : kalman_song.m.....	C-25
C.3.10. Sous-programme : kalman_song_salgado.m.....	C-27
C.3.11. Sous-programme : rls_simple.....	C-30
C.3.12. Sous-programme : rls_lambda.....	C-31
C.3.13. Sous-programme : rls_reset.m.....	C-33
C.3.14. Sous-programme : rls_salgado.....	C-35
C.3.15. Sous-Programme : rls_IMCEVI.m.....	C-37
Annexe D : Compléments sur la schématique.....	D-1
D.1. ÉTAGE AMPLIFICATEUR.....	D-1
D.2. GAIN VARIABLE.....	D-2
Annexe E : Description du micro-contrôleur 80C196KD20.....	E-1
E.1. LES PRINCIPALES CARACTERISTIQUES DU MICRO-CONTROLEUR.....	E-1
E.2. COMPOSITION INTERNE.....	E-2
E.3. LES PRINCIPALES FONCTIONS.....	E-3
E.3.1. Les événements internes.....	E-3

E.3.2. Les périphériques internes.....	E-3
Annexe F : Programme assembleur du micro-contrôleur	F-1
Annexe G : Programme assembleur du module de mesure	G-1
Annexe H : Documentation sur les transducteurs utilisés	H-1

Liste des figures

Figure 2-1 : Schéma général d'un système de mesure [Barwicz96].....	11
Figure 2-2 : Modèle de système de mesure [Barwicz96]	12
Figure 2-3 : Rapport de réflexion	14
Figure 2-4 : Réflexion d'une onde sur un mur.....	15
Figure 2-5 : Réflexion 3D.....	15
Figure 2-6 : Simple réflexion.....	16
Figure 2-7 : Multiple réflexion	16
Figure 2-8 : Aucune réflexion.....	16
Figure 2-9 : Atténuation de l'onde pour le transducteur Polaroid[Polaroid83]	23
Figure 2-10 : Les champs acoustiques.....	24
Figure 2-11 : Schéma bloc du transmetteur [Bentley95]	24
Figure 2-12 : Schéma bloc du récepteur [Bentley95].....	25
Figure 2-13 : Modèle mécanique du transducteur [Frey69]	26
Figure 2-14 : Modèle électrique de la structure mécanique [Frey69].....	26
Figure 2-15 : Principe de la mesure par temps de vol	27
Figure 2-16 : Méthode de différence de phase	29
Figure 2-17 : Bloc conversion A_2/A_3	32
Figure 3-1 : Écho mesuré provenant d'un cylindre	39
Figure 3-2 : Influence de l'échantillonnage sur le résultat de l'intercorrélation	42
Figure 3-3 : Influence du 2eme echo sur l'intercorrélation	43
Figure 3-4 : Organigramme du programme pour la mesure de distance	44
Figure 3-5 : Représentation du système sensoriel de mesure de position	45
Figure 3-6 : Organigramme de l'algorithme d'estimation des paramètres de la droite	47
Figure 3-7 : Réflexion sur un mur	48
Figure 3-8 : Système de mesure à double transducteur	49
Figure 3-9 : Géométrie du système à double transducteur	49

Figure 3-10 : Le repère mobile	50
Figure 3-11 : Générateur de bruit	61
Figure 3-12 : Estimations à l'aide du filtre de Kalman étendu.....	62
Figure 3-13 : RLS avec ou sans facteur d'oubli	65
Figure 3-14 : RLS avec facteurs d'oubli et resetting exponentiel.....	67
Figure 3-15 : RLS_IMCEVI appliqué à l'estimation des coefficients de la tangente.....	69
Figure 3-16 : Détails de l'estimation avec le RLS_IMCEVI	70
Figure 4-1 : Schéma fonctionnel de l'excitation	73
Figure 4-2 : Schéma fonctionnel de la conversion	73
Figure 4-3 : Schéma d'excitation (d'après Polaroid)	74
Figure 4-4 : Filtre sélectif	76
Figure 4-5 : Interface simplifiée du 87C196KD	83
Figure 4-6 : Schéma bloc du HSO [Intel92]	84
Figure 4-7 : Schéma bloc du convertisseur A/N [Intel92]	85
Figure 4-8 : Schéma bloc du HSI [Intel92].....	87
Figure 5-1 : Atténuation de l'écho avec le gain variable.....	99
Figure 5-2 : Amplification de l'écho avec le gain variable	100
Figure 5-3 : Masquage de l'écho avec le gain variable	101
Figure 5-4 : Niveau de bruit du système d'acquisition	101
Figure 5-5 : Schéma de test.....	102
Figure 5-6 : Atténuation versus l'alignement.....	103
Figure 6-1 : Répartition des groupes.....	107
Figure 6-2 : Composition d'un groupe	108
Figure 6-3 : Adressage des modules	108
Figure 6-4 : Composition d'un demi-groupe.....	109
Figure 6-5 : Liaison dite d'instrumentation.....	110
Figure 6-6 : Structure IEEE 1451 [EDC98].....	112
Figure 6-7 : Organisation structurelle d'un esclave	117
Figure 6-8 : Protocole de communication du maître	118
Figure 6-9 : Protocole de communication des esclaves.....	119

Figure 6-10 : Organisation structurelle du maître.....	120
Figure A-1 : Relation inverse.....	A-1
Figure A-2 : Modèle électrique en transmission [Bentley95]	A-2
Figure A-3 : Relation directe	A-3
Figure A-4 : Modèle électrique en réception [Bentley95]	A-3
Figure A-5 : Transmission-Réception acoustique	A-4
Figure A-6 : Atténuation versus la fréquence.....	A-7
Figure A-7 : Schéma bloc de la sensibilité totale	A-10
Figure B-1 : Influence du nombre de périodes émises	B-3
Figure C-1 : Changement de repère.....	C-1
Figure C-2 : Transformation géométrique pour le simulateur.....	C-3
Figure C-3 : Déduction de la trajectoire par permutations	C-4
Figure C-4 : Positionnement des transducteurs	C-5
Figure C-5 : Mesure de distance transducteur-mur	C-6
Figure C-6 : Paramètres et leur environnement.....	C-8
Figure D-1 : Interface micro-contrôleur avec le bloc excitation	D-1
Figure D-2 : Gain variable par réseau R/2R	D-3
Figure E-1 : Schéma bloc interne du 8XC196KD [Intel92]	E-2
Figure E-2 : Schéma bloc du cœur du 8XC196KD [Intel92]	E-2
Figure F-1 : Parallélisme dans l'acquisition.....	F-2
Figure F-2 : Organigramme d'acquisition (1/2)	F-3
Figure F-3 : Organigramme d'acquisition (2/2)	F-4
Figure F-4 : Organigramme pour le calcul du temps de vol (1/2)	F-5
Figure F-5 : Organigramme pour le calcul du temps de vol (2/2)	F-6

Liste des tableaux

Tableau 1-1 : Principales caractéristiques des éléments sensoriels	3
Tableau 2-1 : Atténuation par rapport à la fréquence et l'humidité	19
Tableau 2-2 : Equivalence des grandeurs acoustiques, mécaniques, électriques [Frey69] ..	26
Tableau 2-3 : Correspondance entre l'intervalle temporel et le gain de l'amplificateur	32
Tableau 3-1 : Résultats de la méthode avec sinus échantillonné à 3/4 de T	40
Tableau 3-2 : Résultats de la méthode avec fonction porte	40
Tableau 3-3: Paramètres de simulation de la figure 3-12	62
Tableau 3-4: Paramètres de simulation de la figure 3-13	64
Tableau 3-5 : Paramètres de simulation de la figure 3-14	66
Tableau 3-6 : Paramètres de simulation de la figure 5-15	69
Tableau 4-1 : Combinaison des commandes des interrupteurs versus le gain.....	77
Tableau 5-1 : Mesure du temps de vol.....	97
Tableau 5-2 : Répartition des mesures.....	98
Tableau 5-3 : Influence de l'alignement.....	103

Liste des symboles et abréviations

- $\hat{\bullet}$: valeur estimée de
- $\bar{\bullet}$: valeur moyenne de
- \bullet : valeur exacte de
- $\tilde{\bullet}$: variable bruitée de
- \bullet^T : transposée d'une matrice ou d'un vecteur
- \bullet^{-1} : inversion d'une matrice ou d'un vecteur

- a : rayon du transducteur (en mètre)
- \mathbf{a} : vecteur des estimés
- A : amplitude de signal (en volt)
- A : signal analogique
- \mathbf{A} : matrice de transition
- \mathbf{a}_0 : vecteur des estimés servant à la linéarisation
- abs : diminutif de valeur absolue
- ACHi : canal i de conversion A/N
- AD_COMMAND : registre 8 bits pour le déclenchement du convertisseur A/N
- AD_RESUT : registre 16 bits contenant le résultat de conversion et son état
- AD_TIME : registre 8 bits (CAM+SAM) contenant la configuration du convertisseur A/N
- AD7224 : circuit intégré comprenant un convertisseur A/N 8bits
- ADG333A : circuit intégré avec un quadruple interrupteur analogique du type SPDT
- ANGNDi : masse analogique du canal i de conversion A/N
- AM : modulation d'amplitude, *Amplitude Modulation*
- A/N : convertisseur analogique à numérique
- \mathbf{b} : vecteur de mesure
- biais: indique le niveau pour déclencher la réinitialisation de l'algorithme

- \mathbf{b}_0 : vecteur de mesure servant à la linéarisation
 \mathbf{B} : vecteur d'état d'entrée
 C : condensateur (en farad)
 \mathbf{C} : vecteur de mesure
 C_a : vitesse acoustique, appelée célérité (en m.s^{-1})
 C_{ai} : vitesse acoustique, appelée célérité, dans le milieu i
 CAM : structure matérielle qui gère les événements, *Content Access Memory*
 C_{amb} : célérité dans l'air ambiant ou à 20°C (en m.s^{-1})
 C_{gh} : résultat de l'intercorrélation entre les variables g et h
 CMOS : standard logique, *Complementary Metal-Oxide Semiconductor*
 CONV : mot de 5 bits contenant le temps de conversion de convertisseur A/N
 d : distance (en mètre)
 d_i : distance donnée par le transducteur d'index i
 \mathbf{D} : Dénominateur de la fonction $f(a,b)$
 DSP : Processeur des signaux dédié, *Digital Signal Processing*
 d_r : rapport de sensibilité du transducteur
 dt : dérivée temporelle
 dx : distance en $[0x)$ entre le repère mobile et le transducteur
 dy : distance en $[0y)$ entre le repère mobile et le transducteur
 $E[\bullet]$: espérance mathématique de
 EEPROM : ROM effaçable électriquement, *Electrically Erasable Programmable ROM*
 $f(\bullet)$: fonction de
 F : force (en N)
 f_{acq} : fréquence d'acquisition (en Hz)
 facteur_bruit : multiplicateur du niveau de bruit
 f_e : fréquence d'émission (en Hz)
 F_{IN} : force en entrée
 F_{OUT} : force en sortie
 g : variable quelconque

h : variable quelconque
 H : taux d'humidité relative (%)
 $HSIO$: unité de sorties et d'entrées hautes vitesses, *High Speed Input/Output*
 HSI : unité d'entrées hautes vitesses, *High Speed Input*
 $HSI_COMMAND$: registre 8 bits de configuration du HSI
 HSI_STATUS : registre 8 bits contenant l'état du HSI
 HSI_TIME : registre 16 bits contenant le temps de l'événement HSI
 HSO : unité de sorties hautes vitesses, *High Speed Output*
 HSO_TIME : registre 16 bits de configuration temporelle du HSO
 $HSO_COMMAND$: registre 8 bits de configuration du HSO
 $HSOTIM$: variable, durée maximum de la plage d'acquisition ultrasonore
 i : index de boucle
 $immu$: indique le nombre de mesure minimal entre deux réinitialisations de l'algorithme
 I : intensité
 I_0 : intensité initiale
 I_0 : intensité ultrasonore initiale
 $INTii$: Type ii d'interruption
 I/O : entrées/sorties, *Input/Output*
 $IOCi$: registre 8 bits de configuration des I/O, *Input/Output Control Register i*
 j : index de boucle
 $J()$: fonction de Bessel de radiation
 k : nombre d'ondes
 k : échantillon
 K : vecteur gain
 k_c : facteur de conversion
 k_m : constante molaire du gaz parfait
 k_r : coefficient de rigidité du cristal (du transducteur)
 K_r : paramètre de rigidité du cristal
 k_θ : rapport de la chaleur spécifique à un volume et pression constants

L : inductance (en henry)

L_0 : limite du champ de Fresnel (en mètre)

m : masse (kg)

m : masse molaire (g.mol^{-1})

max : diminutif de maximum

min : diminutif de minimum

MUR160 : diode de permutaion haute fréquence et de puissance (ref : Motorola)

n : variable quelconque

nb_ité : nombre d'itération

N : nombre de cycles

N : signal numérique

N/A : convertisseur numérique à analogique

o : variable quelconque

Obs : matrice d'observabilité

OTPROM : ROM écrivable une seule fois, *Only-Time Programmable Read Only Memory*

p : coefficient de l'équation de la tangente

P : pression atmosphérique (en Pa)

P : matrice de la covariance de l'erreur

P_0 : pression acoustique à 1 mètre (en Pa)

P_0 : matrice initiale de la covariance de l'erreur

P_a : pression à l'émission (aller) (en Pa)

P_{ar} : pression totale (aller-retour) (en Pa)

pcqs : produit de p par le cosinus et de q par le sinus

$P_{emis(x)}$: pression émise à une distance x

P_r : pression à la réception (retour) (en Pa)

$P_{refl(x)}$: pression réfléchie à une distance x

P_s : pression spécifique de l'air (en Pa)

PTS : unité de traitement des interruptions, *Peripheral Transaction Server*

PWM : générateur de rapport cyclique, *Pulse Width Modulator*

q : charge électrique (en coulomb)

q : coefficient de l'équation de la tangente
Q : matrice d'autocorrélation du bruit d'entrée
r : impédance acoustique (en rayl)
R : résistance (en ohm)
R : matrice d'autocorrélation du bruit de mesure
R : problème de reconstitution
R/2R : réseau de résistances utilisé dans les convertisseurs N/A
R_a : rapport de réflexion
RAM : mémoire vive, *Random Access Memory*
R_L : résistance de charge
RLS : algorithme à moindres carrés récursif, *Recursive Least Square*
ROM : mémoire morte, *Read-Only Memory*
S : surface
s_d : distance entre les 2 transducteurs (en cm)
SAM : mot de 3 bits contenant le temps d'échantillonnage du convertisseur A/N
seuil : indique le rapport de déclenchement de la réinitialisation de l'algorithme
signe(u) : fonction signe de la variable u
SN28784 : Circuit analogique dédié au kit Polaroid (ref : Texas Instruments)
SPDT : interrupteur avec 2 positions et une commande (*Single Pole Double Throw*)
SPL : niveau de pression acoustique, *Sound Pressure Level*
SRAM : mémoire vive statique, *Static Random Access Memory*
t : temps (en s)
t_{reset} : valeur entre deux réinitialisations automatiques
tacq : temps correspondant à la mémorisation de l'écho
T_c : température (en degré Celcius)
T_e : période d'échantillonnage (en s)
Tempo : diminutif de temporisation
T_k : température (en degré Kelvin)
T_i : transducteur d'indice i
TL : pertes en transmission, *Transmitting Losses*

Tof : temps de vol (en s)
 TL851 : circuit numérique dédié au kit Polaroid (ref : Texas Instruments)
 TL852 : circuit analogique dédié au kit Polaroid (ref : Texas Instruments)
 TPU : unité de temps programmable, *Time Process Unit*
 T_{ri} : transistor d'indice i
 TTL : standard logique, *Transistor-Transistor Logic*
 u : déformation
 u : signal d'excitation
 u : variable quelconque
 U : vecteur d'entrée
 U_L : tension aux bornes de l'inductance
 v : vecteur des quantités influentes
 v : vecteur des grandeurs d'influence
 v : bruit de mesure
 V : tension (en volt)
 Variance : variance d'un vecteur
 V_{recu} : tension reçue
 V_{emis} : tension émise
 V_{OUT} : tension de sortie
 $VREF_i$: tension de référence du canal i de conversion A/N
 V_s : tension de source
 w : pulsation (en rad.s^{-1})
 w : bruit d'entrée
 x : déplacement selon l'axe [Ox)
 x : grandeur à mesurer ou le mesurande
 x : vecteur des mesurandes
 x : vecteur d'état
 X : vecteur d'état
 \dot{X} : dérivée du vecteur d'état

- x_i : abscisse du point i
 x_M : abscisse de l'objet dans le repère mobile
 $XMIT$: train d'ondes pour l'excitation du transducteur
 x_P : abscisse de l'objet dans le repère global
 x_P : abscisse du repère mobile dans le repère global
 y_M : ordonnée de l'objet dans le repère mobile
 y_P : ordonnée de l'objet dans le repère global
 y : déplacement selon l'axe [0y)
 y : vecteur de résultats de conversion
 y : variable de sortie du système d'état
 Y : vecteur de sortie
 y_i : ordonnée du point i
 y_P : ordonnée du repère mobile dans le repère global
 z : déplacement selon l'axe [0z)
 Z : impédance (en ohm)
 Z_G : impédance du générateur
 Z_{IN} : impédance d'entrée
 z_M : coordonnée en [0z) du repère mobile
 Z_{OUT} : impédance de sortie
 z_P : coordonnée en [0z) du repère global
 68HC11 : microcontrôleur 8 bits (ref : Motorola)
 68HC12 : microcontrôleur 16 bits entièrement compatible 68HC11 (ref : Motorola)
 68HC16 : microcontrôleur 16 bits (ref : Motorola)
 8XC196 : microcontrôleur 16 bits (ref : Intel)
- α : angle entre la normale de la surface et la direction des ondes
 α : coefficient d'oubli pour le vecteur gain, K
 α_1 : angle d'incidence dans le milieu 1
 α_2 : angle d'incidence dans le milieu 2

α_{us} : coefficient d'amortissement du milieu (en dB/m)

β : angle d'ouverture

β : coefficient d'oubli qui limite la plus petite valeur propre de P

γ : angle d'inclinaison du mur

γ : variable interne pour les coefficients d'oubli

δ : coefficient d'oubli qui limite la plus grande valeur propre de P

ϕ : différence de phase (en degré)

ϕ : coefficient d'oubli exponentiel

ϕ_{360} : différence de phase modulo 360° (en degré)

θ : angle

θ_0 : angle d'ouverture

θ_c : angle critique

ρ_i : masse volumique dans le milieu i

τ : constante de temps

λ : longueur d'onde ($m.s^{-1}$)

λ : facteur d'oubli

λ_m : résistance mécanique du système (kg/s)

σ : variable interne pour les coefficients d'oubli

υ : variable interne pour les coefficients d'oubli

ξ : facteur d'amortissement

$\{\tilde{y}_n\}$: séquence bruitée du vecteur y qui est le résultat brut de mesure

$\{\tilde{u}_n\}$: séquence du signal qui excite la mesure

Chapitre 1 :

Introduction

1.1 Contexte

Ce travail présente une méthode de localisation des objets dans un environnement quelconque à partir de simples mesures de distance. Ce positionnement permet à un système sensoriel de donner des informations sur l'environnement proche de façon à éviter tout contact entre le système mobile et les objets.

Alain Pruski introduit le problème de la planification de trajectoire sous la forme suivante [Pruski96]:

« La capacité d'autonomie d'un robot est directement liée à la puissance d'appréhension de l'environnement »

« La chaîne de réaction se compose de plusieurs modules : la perception d'événement, la modélisation, la détermination du type d'événement, la prise de décision »

« La perception d'événement constitue le rôle du capteur. Il va traduire un état physique en signaux électriques. La modélisation constitue à associer une représentation informatique ou/et mathématique aux informations délivrées par le capteur, ceci dans le but de pouvoir les manipuler avec des outils disponibles. La détermination du type d'événement consiste à trouver, à partir du modèle ou éventuellement de plusieurs modèles, l'événement qui vient d'être perçu. »

Ces trois situations résument bien le contexte du travail. Ce dernier va traiter de la perception, la modélisation et la détermination du type d'événement. La prise de décision ne fait pas l'objet de ce mémoire.

Les éléments sensoriels, dans un système mobile, peuvent avoir deux rôles : (i) ils délivrent une information caractérisant l'environnement et sont appelés extéroceptifs (mesure sans contact) ; (ii) ils délivrent une information sur le fonctionnement du système, ils sont appelés proprioceptifs. Pour la perception, seulement la première catégorie est utilisée, la deuxième intervient dans la localisation. Pour chaque catégorie, on retrouve deux modes de capteurs : les capteurs actifs ou passifs. Les capteurs passifs mesurent une énergie entièrement fournie par le phénomène mesuré (exemple les capteurs de proximité de type inductif ou capacitif). Les capteurs actifs fournissent une énergie au milieu pour obtenir des mesures (exemple: les télémètres à laser ou à ultrason). On peut définir le type d'élément sensoriel voulu suivant le cahier des charges suivant :

- étendue de mesure allant jusqu'à plusieurs mètres,
- résolution sous le centimètre
- prix: le plus faible
- traitement simple
- variation aux conditions atmosphériques : faible

Pour justifier le choix futur, le tableau 1-1, adapté de [IRSST95], donne les principales caractéristiques des éléments sensoriels qui mesurent la distance.

Les détecteurs de proximité (capacitif, inductif, piézo-électrique, magnétique...) ont une portée trop faible qui n'atteint pas le mètre. De ce fait, ils sont éliminés. Les capteurs infrarouges actifs ont une portée de plusieurs mètres mais l'amplitude mesurée par le récepteur est une fonction non linéaire entre la distance et la couleur de l'objet. Les radars et surtout les lasers présentent une très bonne précision et étendue de mesure, mais leurs prix sont très élevés. Il reste dans une gamme de prix abordables: les caméras 2D et les télémètres ultrasoniques, les caméras ont le grave défaut de demander un traitement important et d'avoir une sensibilité grande aux variations de lumière. Les capteurs ultrasonores sont choisis comme moyen de mesurer de la distance.

Tableau 1-1 : Principales caractéristiques des éléments sensoriels

	capteur infrarouge	télémètre à laser	radar	télémètre à ultrason	détecteurs de proximité	vision 2D et 3D
actif/passif	actif	actif	actif	actif	passif et actif	passif et actif
mesure	luminosité	Distance	distance	distance	présence	luminosité
principe	détection ondes infrarouges	Tof ¹ , modulation AM	Tof ¹ , effet Doppler	Tof ¹	plusieurs	imagerie triangulation
échelles	10 ⁻² à 10 ¹ mètres	0.2 à 10 ³ mètres	1 à 10 ³ mètres	10 ⁻² à 10 ² mètres	10 ⁻³ à 1 mètre	10 ¹ mètres
précision, résolution	faible à bonne	très élevée	bonne à élevée	faible	faible à très élevée	bonne à élevée
fréquence acquisition	très élevée	très élevée	très élevée	faible	très élevée	faible à élevée
fiabilité	bonne	dépend des traitements	dépend des traitements	faible à bonne	élevée	dépend des traitements
résistance mécanique	bonne à élevée	faible à bonne	bonne	très bonne	très bonne	bonne
insensibilité aux variations climatiques	bonne	faible	bonne	moyenne	faible à élevée	faible
coût	très faible	élevé	très élevé	très faible	très faible à élevé	faible (2D) à très élevé
dimensions	faibles	faibles	élevées	faibles	faibles	moyennes

¹ Temps de vol en anglais (*Time of flight*)

1.2 Problématique

La problématique est ici double, contourner les problèmes liés aux réflexions spéculaires des ondes ultrasonores et évaluer la position à l'aide de distances trouvées avec l'unité sensorielle ultrasonore.

Pendant la propagation acoustique, l'onde peut se réfléchir en frappant d'autre(s) objet(s), ce qui va fausser le temps de parcours ou bien l'onde peut se disperser dans l'environnement sans jamais revenir au récepteur. La propagation acoustique ressemble à la propagation optique avec les réflexions et les dispersions à la seule différence que la propagation acoustique s'effectue très lentement comparée à la vitesse de la lumière.

La localisation des objets détectés par le module sensoriel fait appel à la fusion de données pour estimer les coordonnées de ces objets. Cette fusion est basée sur la disposition des éléments sensoriels et un traitement algorithmique est nécessaire pour évaluer la position. Il faut que l'algorithme présente une certaine immunité aux bruits de mesure pour pouvoir donner des résultats corrects.

1.3 Objectifs

L'objectif principal du mémoire est de proposer un système de localisation de l'environnement proche d'un système mobile. Ce système utilise uniquement comme source sensorielle les capteurs ultrasonores. Il va permettre de donner des mesures sur le positionnement des objets avoisinants. Pour déterminer la position d'un objet, deux informations sont nécessaires: la distance et l'orientation. Ces deux informations vont constituer les deux sous-objectifs du mémoire. Le premier sous-objectif consiste à réaliser un module de mesure de distance à partir d'un transducteur ultrasonore. Ce module simple, basé sur une structure matérielle et logicielle, devra être fiable et précis. Une étude approfondie de l'algorithme retenu est effectuée pour déterminer les meilleures conditions de fonctionnement vis à vis des sources influentes. Le deuxième sous-objectif est de simuler un algorithme de fusion de données pour obtenir la position de l'objet à partir des distances mesurées. Le positionnement d'un objet est donné, ici, par son équation de la tangente au point de mesure. L'algorithme qui sera développé, estimera les coefficients de l'équation de la tangente de l'objet détecté et devra être robuste aux bruits de mesure. Une stratégie de placement des transducteurs comprenant une partie interface est élaborée.

1.4 Les méthodes existantes

Le positionnement de système mobile est essentiellement traité en robotique. On divisera cette étude en deux parties correspondant aux 2 sous-objectifs. Auparavant, un rappel sur la classification des méthodes de positionnement est proposé.

On peut répertorier 3 grandes méthodes de positionnement d'un système mobile :

- celle qui utilise uniquement des données odométriques qui proviennent uniquement du déplacement de la base mobile, il s'agit d'un système dit "aveugle", car il suppose qu'aucun élément ne puisse le gêner. Ce modèle n'utilise aucun moyen externe pour corriger les erreurs qui s'accumuleront;
- celle qui utilise des données externes pour se localiser, on retrouve deux sous-catégories: (i) le système reçoit des informations provenant de balises placées dans son environnement de travail, ce type de méthode est utilisé en industrie pour des tâches répétitives de transport; (ii) il reçoit des données sensibles caractérisant l'environnement proche, cette méthode est souvent utilisée en robotique;
- on retrouve les méthodes hybrides qui regroupent les deux premières catégories.

De même, chaque méthode peut avoir son traitement respectivement hors ligne et en ligne [Pruski96] :

- « En attendant que suffisamment de mesures soient acquises afin de les analyser globalement ».
- « De traiter les données au fur et à mesure de leur présence afin d'affiner progressivement l'estimé de la valeur réelle ».

Pour la réalisation du module de mesure, deux techniques sont possibles [Kocis96] : la mesure du temps de vol de l'onde et la mesure de la différence de phase. La mesure du temps de vol, en anglais *Time of flight* (Tof), compte un nombre de cycles entre le début du signal émis et celui reçu. Cette méthode est simple à réaliser, il s'agit d'un détecteur de niveau qui s'active à l'émission et qui s'arrête au premier écho reçu; la distance est alors proportionnelle au temps de vol pendant lequel le détecteur est actif [Ervett95]. Cette méthode est très sensible aux bruits de mesure (impulsions) qui peuvent arrêter le compteur. La deuxième méthode est utilisée pour les très petites distances, et consiste à mesurer la différence de phase entre l'onde émise et l'onde reçue. La phase est proportionnelle à la distance. Pour des distances supérieures à la longueur d'onde, on peut combiner la méthode de la différence de phase avec celle du temps de vol pour augmenter la résolution du résultat. Cette méthode utilise des ressources d'acquisition et de mémorisation importantes puisqu'il faut trouver la phase (reconstruction numérique du signal) [Scharft96]. Une alternative est d'utiliser des méthodes de filtrages adaptés pour déterminer la fin du temps de vol (début de l'écho).

Pour déterminer le positionnement de l'objet détecté par le module de mesure, de nombreuses méthodes existent, on peut décomposer ces méthodes en deux classes: celle qui positionne l'objet dans une grille; celle qui recherche les coordonnées exactes de l'objet.

La première classe attribue un 0 ou un 1 à chaque case de la grille pour chaque mesure, le 0 est donné si l'objet ne se trouve pas dans cette case et le 1 correspond à une possibilité de la présence de l'objet dans cette case [Matthies88]. Dans le cas d'un environnement qui bouge (environnement fixe et le système de mesure se déplace ou vis versa) on met à jour la grille en additionnant les nouvelles valeurs avec les anciennes [Borenstein91], [Horn97].

La deuxième classe regroupe les méthodes d'estimation mathématique, ces méthodes estiment la position de l'objet de façon directe (coordonnées) ou indirecte (exemple : équation de la tangente détectée). Cette classe peut être décomposée en trois catégories pour la recherche de la localisation de l'objet détecté à l'aide de plusieurs éléments sensitifs ultrasonores :

- Les méthodes qui utilisent uniquement la géométrie pour la localisation.
- Les méthodes qui estiment les paramètres de l'environnement par filtrage.
- Les méthodes qui utilisent des moyens probabilistes pour déterminer le type d'environnement.

Le premier groupe de méthodes est basé sur l'étude de la propagation des ondes et la configuration des transducteurs. L'une des principales préoccupations est de déterminer le type de l'environnement (coin, plan ...). Cette détermination est basée sur les paramètres électriques des ondes retournées en plus de la configuration des transducteurs. Les principaux travaux sont ceux de [Kuc92], [Kleeman95] et [Barshan90]. On remarquera que ces auteurs sont ou ont été au laboratoire *Intelligent sensors laboratory* de l'université de Yale. Une autre stratégie est l'utilisation d'un concept de capteur virtuel avec [Hanebeck94].

Le deuxième groupe de méthodes est basé principalement sur le filtre de Kalman. On retrouve les travaux de Durrant-Whyte et Leonard qui ont participé à rendre cette méthode populaire [Leonard92]. De nombreux auteurs ont effectué des recherches sur cet algorithme, on retrouve Sabatini qui positionne l'objet détecté et estime son rayon [Sabatini94], Curran qui utilise l'algorithme dans un environnement perpendiculaire tel un labyrinthe [Curran93], Song qui l'utilise pour estimer la tangente de l'objet détecté [Song96], Chong qui utilise pour la localisation une variante du filtre de Kalman, appelée JUKL [Chong97]. On peut citer quelques travaux utilisant la minimisation de l'erreur quadratique comme décrit dans [Ko96].

Le dernier groupe donne des informations sur le type de l'environnement à l'aide des méthodes de votes ou de Dempster-Shafer [Ayrulu97], ou probabiliste avec celle de Bayes [Fatemi96], ou avec l'estimation du maximum de ressemblance [Peremans93] et [Arikan95].

Cependant d'autres méthodes ont aussi été évaluées comme l'emploi des réseaux de neurones [Song93] et [Ferraris93], la logique floue [Baglio96] et des réseaux ultrasonores basés sur la différence de phase [Webb96] et [Shraft96].

1.5 Méthodologie

La méthodologie utilisée consiste premièrement par une étude bibliographique sur les méthodes permettant la perception d'environnement par capteurs ultrasonores. Cette étude cernerait la méthode la plus apte à effectuer le positionnement des objets détectés. Comme le mémoire se divise en deux sous-objectifs, chaque objectif aura son algorithme dédié qui devra présenter une bonne immunité au bruit de mesure. Les algorithmes seront évalués à l'aide des données synthétiques qui modélisent l'environnement, le transducteur, la propagation. Une phase d'amélioration des paramètres de l'algorithme est prévue. Pour la partie matérielle chargée de la mesure de distance, une solution compacte et simple répondant aux types des données voulues par l'algorithme de positionnement sera conçue et sera expérimentée.

1.6 Organisation du mémoire

Le mémoire se divise comme suit.

Dans le chapitre deux, une présentation du système de mesure ultrasonore est décrite, elle consiste en la définition du système de mesure, la théorie sur la propagation acoustique touchant à la problématique, les moyens d'évaluer la distance et pour finir la composition du système de mesure ultrasonore envisagé.

Le troisième chapitre traite des algorithmes pour les deux reconstitutions. Dans un premier temps, on étudie la reconstitution de la distance à l'aide de la méthode du temps de vol. Dans la deuxième partie, on étudie la reconstitution de la position pour connaître les coefficients du segment de droite qui modélise l'objet détecté. Pour chaque partie, des simulations sont effectuées avec des données synthétiques qui sont créées.

Le quatrième chapitre traite des problèmes de conception pour implanter le système de mesure ultrasonore qui estime la distance de l'objet détecté. On peut discerner la conception en deux grandes parties qui sont les deux blocs du système de mesure. La partie conversion traite de l'interface analogique entre la structure numérique où se trouve l'algorithme et le transducteur. La partie reconstitution définit un cahier des charges sur le choix de la structure numérique et intègre la reconstitution dans cette structure. Pour finir ce chapitre, une réflexion est faite sur la structure matérielle pour la reconstitution de position.

Le chapitre cinq est consacré aux résultats expérimentaux, on étudie les résultats de chaque partie structurelle, et on propose une méthode pour tester le système complet.

Le chapitre six explique l'élaboration d'une stratégie de placement des transducteurs comprenant leur interface avec la structure matérielle.

Le mémoire se termine par une conclusion qui fait le bilan du travail accompli et est suivi de nombreuses annexes qui apportent des détails sur les choix effectués dans ce travail.

Chapitre 2 :

Système de mesure ultrasonore

Ce chapitre contient les différents éléments nécessaires pour la compréhension du système de mesure ultrasonore. Une présentation générale d'un système de mesure est faite suivie d'une description théorique de la propagation acoustique de façon à comprendre les blocs fonctionnels nécessaires à la mesure de la distance à l'aide d'un transducteur ultrasonore. Ce dernier, un élément mécanique, va être modélisé sous forme de composants électroniques afin d'obtenir son équation de transfert pour pouvoir simuler par la suite sa réponse.

2.1 Généralités

"La modélisation des systèmes de mesure a pour but de donner une réponse à la question fondamentale: De quelle façon la représentation d'une grandeur fournie par un instrument ressemble-t-elle à cette grandeur elle-même? Pour répondre à cette question, les systèmes de mesure sont en général décomposés en un certain nombre d'éléments. Les performances de chaque élément sont décrites au moyen d'un modèle physique adéquat et mises sous forme d'une caractéristique de transfert." [BenSlima93]

Sous la forme la plus simple, la chaîne de mesure peut donc être réduite à un capteur et son circuit conditionneur éventuel associé à un appareil indicatif. En réalité, ce modèle simple n'est pas toujours utilisable pour avoir une exploitation suffisante des données, cette nécessité est due essentiellement à des paramètres environnementaux et des non-linéarités des caractéristiques.

Une nouvelle approche constituée de deux blocs fondamentaux, est proposée par [Barwicz96]. Le premier effectue la conversion de la grandeur à mesurer en une grandeur discrète, le deuxième se nomme la reconstitution, c'est à dire la correction des imperfections du modèle.

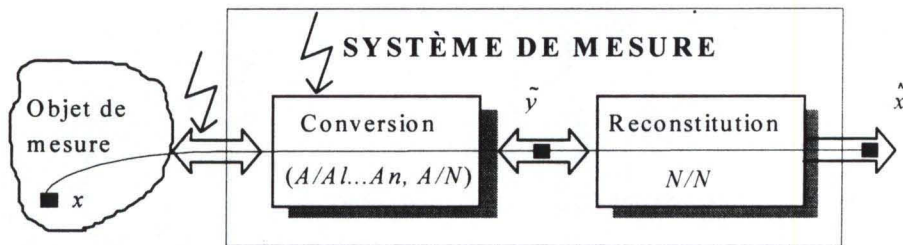


Figure 2-1 : Schéma général d'un système de mesure [Barwicz96]

où x : La grandeur à mesurer (la mesurande)

\tilde{y} : Le résultat de conversion

\hat{x} : Le résultat de mesure (résultat de reconstitution)

La conversion est une transformation du signal de mesure x (grandeur à mesurer) en un signal y , qui représente x dans un domaine plus approprié à l'utilisateur, c'est à dire sous forme numérique.

La reconstitution consiste à l'estimation du signal x en se basant sur le résultat de conversion y .

Chaque processus de mesure est perturbé par la présence de grandeurs d'influences externes et internes qui affectent non seulement l'acquisition de l'information mais aussi sa transmission. Pour écrire le problème de reconstitution, R , on a recours à un modèle mathématique qui relie le signal y au signal x . Un tel modèle s'écrit sous la forme [Barwicz96]

$$y = R [x ; v] \quad (2-1)$$

où \mathbf{y} : le vecteur de résultats de conversion

\mathbf{v} : le vecteur de grandeur d'influence

\mathbf{x} : le vecteur des mesurandes

Mais en allant plus loin dans la philosophie des systèmes de mesure, on peut détailler, la précédente figure par celle proposée par Morawski [Barwicz96].

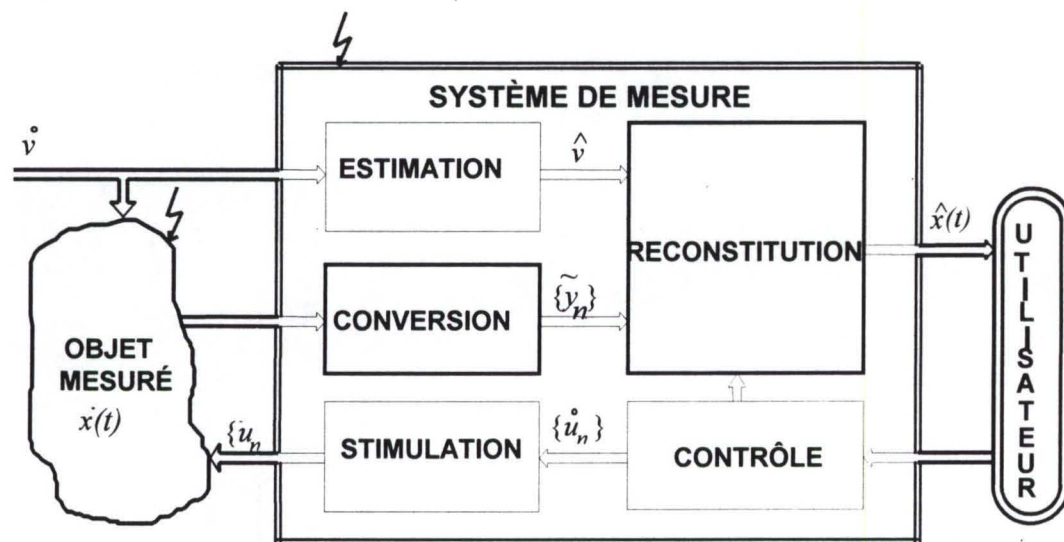


Figure 2-2 : Modèle de système de mesure [Barwicz96]

où : $\hat{\mathbf{v}}$ Le vecteur estimé des quantités influentes

$\{\tilde{\mathbf{y}}_n\}$ Une séquence bruitée du vecteur \mathbf{y} qui est le résultat brut de mesure

$\{\tilde{\mathbf{u}}_n\}$ Une séquence du signal qui excite la mesure

On remarquera que les deux blocs en traits gras se retrouvent toujours dans un système de mesure, les blocs en traits minces peuvent exister ou pas en fonction de l'application.

Une définition de chaque bloc est disponible dans [Barwicz96]. Dans la section 2-5 de ce chapitre, ces blocs sont décrits pour la mesure de distance par ultrasons.

2.2 Propagation ultrasonore

Le principe de la propagation ultrasonore repose sur les propriétés acoustiques, c'est à dire la propagation du son. La plage de fréquences se trouve au-delà de la limite auditive par l'homme, on situe la plage de fréquences à partir de 18 kHz. La vitesse du son est d'environ de 340 m/s et est relativement lente comparée à la célérité de la lumière: $3 \cdot 10^8$ m/s. Une grande partie des informations de cette partie provient du livre *Engineering applications of acoustics* [Kleepe82].

2.2.1 La réflexion

Le principe de la réflexion est le même que pour l'optique. Mais la propagation acoustique est une propagation sphérique ou en 3 dimensions. Le cas d'une onde plane monofréquence est donné dans les chapitres suivants.

Le rapport de réflexion R_a , illustré par la figure 2-3, entre un milieu 1 et 2 est donné en (2-2) . Ce rapport peut être écrit en terme d'angle en (2-3).

$$R_a = \left(\frac{\rho_2 C_{a2} - \rho_1 C_{a1}}{\rho_2 C_{a2} + \rho_1 C_{a1}} \right) \quad (2-2)$$

$$\frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{C_{a1}}{C_{a2}} \quad (2-3)$$

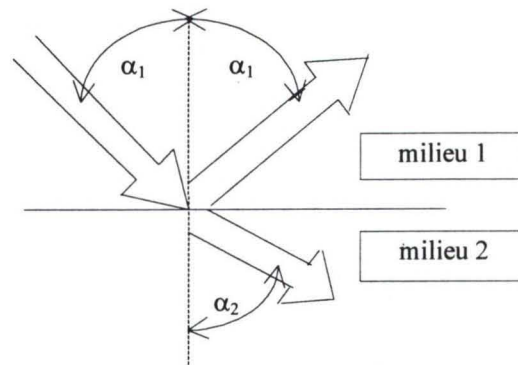


Figure 2-3 : Rapport de réflexion

où ρ_i est la masse volumique

C_{ai} est la vitesse acoustique.

En tenant compte des propriétés des corps, les corps solides ont une célérité du son toujours plus élevée que celle de l'air ambiant. De même, cette caractéristique est vraie pour l'impédance acoustique des solides ($r = C_a * \rho$), l'impédance acoustique est donnée en rayl.

Remarque: Le taux d'intensité réfléchi correspond au carré du coefficient de réflexion.

On peut définir un angle critique θ_c (2-4) qui correspond à l'angle maximum d'incidence où on peut avoir un signal réfléchi.

$$\theta_c = \frac{C_{a1}}{C_{a2}} \quad (2-4)$$

En fonction de la structure de l'objet frappé, les ondes acoustiques présentent une dispersion plus ou moins importante. Lorsque les ondes entrent en collision avec un objet, une partie d'entre-elles est réfléchi de façon symétrique à la normale de la surface de l'objet si ce dernier est lisse et l'autre partie est absorbée par l'objet (une très faible partie pour les objets solides). Les chercheurs ont évalué pour un mur en béton peint, que la quasi-totalité des ondes était réfléchi et le cône de dispersion autour du symétrique à la normale

de l'objet est d'environ 30 degrés (figure 2-4 et 2-5). Avec certains objets en mousse ou molletonnés, le taux de réflexion est très faible.

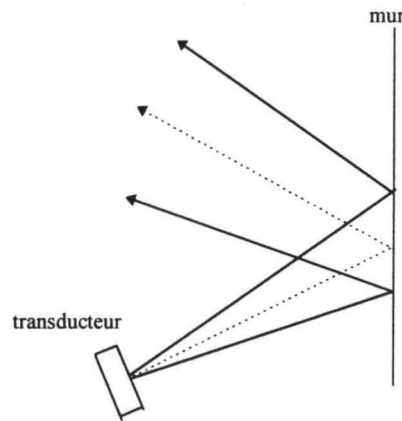


Figure 2-4 : Réflexion d'une onde sur un mur

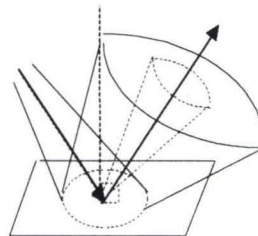


Figure 2-5 : Réflexion 3D

2.2.2 Les réflexions et leur problématique

À partir de cette brève description de la propagation des ondes acoustiques, on comprend la difficulté de gérer les différentes ondes réfléchies, sans perdre de l'information, ni prendre compte des échos possibles. L'information sur l'objet le plus proche se trouvera dans l'onde qui reviendra la première, c'est à dire, celle qui aura parcouru la distance la plus courte. Cependant, il se peut que cette onde provienne du premier objet rencontré (figure 2.6) ou de plusieurs objets, réflexion multiple (figure 2.7). De même manière, il se peut que l'objet rencontré ne renvoie aucune information vers le transducteur, on n'aura pas

d'information sur l'objet non détecté (figure 2.8).

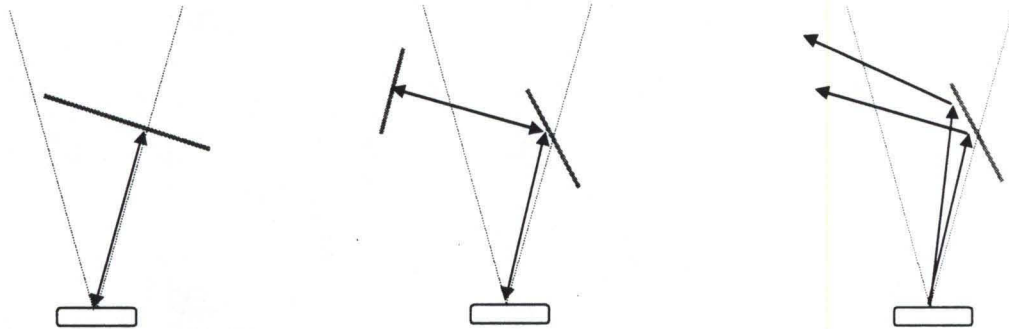


Figure 2-6 : Simple réflexion Figure 2-7 : Multiple réflexion Figure 2-8 : Aucune réflexion

Une autre problématique est de déterminer la position exacte de l'objet dans l'étendue de mesure, avec un seul transducteur comprenant un émetteur et un récepteur intégrés, on peut connaître uniquement la distance à laquelle se trouve l'objet détecté (mesure 1-Dimension). Si on passe à deux transducteurs qui détectent le même objet, on aura une mesure à 2-Dimensions, etc.

2.2.3 Les propriétés acoustiques

La mesure comporte deux étapes : un émetteur convertit un signal électrique en un signal acoustique qui sert de signal de stimulation; un récepteur réalise la conversion inverse. On retrouve ces deux éléments dans un transducteur.

2.2.3.1 La propagation

La vitesse ou la célérité sont deux synonymes. On peut définir la célérité acoustique comme la vitesse dans un gaz parfait dont on connaît ses propriétés (2-5).

$$C_a = \sqrt{\frac{k_\theta \cdot k_m \cdot T_k}{m}} \quad (2-5)$$

avec k_θ : rapport de la chaleur spécifique à un volume et pression constants

k_m : constante molaire du gaz parfait

T_k : température en degré Kelvin

m : masse molaire

Les facteurs d'influence de la vitesse de propagation

La vitesse d'une onde acoustique dans l'air varie en fonction de différents paramètres: la température, l'humidité et sa composition chimique. L'air possède une propriété intéressante de ne quasiment pas modifier la fréquence d'émission du signal émis.

L'influence de la température est donnée en (2-6) pour une humidité normale (37 %):

$$C_a = 20.05\sqrt{T_c + 273.16} \quad (T_c \text{ en } ^\circ\text{C}) \quad (2-6)$$

On peut effectuer une linéarisation de la formule (2-6) pour une plage de -30°C à 30°C

$$C_a = 331.4 + 0.607 \cdot T_c \quad (T_c \text{ en } ^\circ\text{C}) \quad (2-7)$$

On remarquera qu'une variation de température de 5.5°C sur la gamme de -30° à 30° dans l'équation (2-7) provoque une variation de vitesse de 1% .

L'influence de l'hygrométrie est donnée dans l'équation (2-8)

$$C = C_{\text{amb}}(1 + 3,72 \cdot 10^{-5} H) \quad (2-8)$$

avec C_{amb} : célérité dans l'air ambiant (20°C)

H : taux d'humidité relative (%)

L'effet de l'humidité est négligeable (variation maximale de 0.35% à température ambiante).

L'influence des mouvements d'air sur la célérité peut être résolue par la géométrie vectorielle. Par exemple un vent latéral de 10 m.s^{-1} provoque une erreur angulaire de 1.7 degré [= arctan(10 / 343)] cela montre que les mouvements d'air influenceront légèrement la direction de propagation du signal acoustique.

L'influence de la pression ambiante peut varier selon les conditions atmosphériques (anticyclone, dépression et hauteur par rapport au niveau de la mer). Cette dernière est négligeable pour la mesure de vitesse.

En conclusion, la célérité du son dépend principalement de trois facteurs, le premier, la température, affecte plus la mesure que les deux derniers (humidité et pression). Une équation générale (2-9) de la célérité est donnée dans [IRWIN97]

$$C_a = 20,05 \sqrt{\frac{T_c + 273,16}{1 - 3,79 \cdot 10^{-3} \cdot (H.P / P_s)}} \quad (2-9)$$

avec P : pression atmosphérique

P_s : pression de spécifique de l'air qui est donnée dans cet ouvrage.

2.2.3.2 L'atténuation

On étudie actuellement, l'atténuation de l'onde acoustique mais pas l'atténuation entre le signal émis et celui reçu. Pour cette dernière, on multiplie l'atténuation de la pression acoustique qui est calculée ci-dessous, par l'atténuation due à la conversion tension d'excitation vers la pression acoustique émise et par l'atténuation due à la conversion pression acoustique reçue vers la tension. Pour des conditions environnementales stables (pression, température, humidité, fréquence de travail, ...), ces deux fonctions de transfert

sont considérées comme des constantes.

Dans le cas d'une onde plane monofréquence, on définit que l'onde ultrasonore dissipe son énergie dans le milieu qui lui sert de support. Cette atténuation est due aux frottements internes existant dans les corps. Ainsi, l'intensité acoustique (2-10) de l'onde ultrasonore (en W.m^2) va décroître de manière exponentielle en fonction de la distance telle que :

$$I(x) = I_0 e^{-2\alpha_{us}x} \quad (2-10)$$

où x : distance par rapport à la source ultrasonore (en mètre)
 I_0 : intensité ultrasonore initiale
 α_{us} : coefficient d'amortissement du milieu (en décibel par mètre)

Ce coefficient d'atténuation dépend de trois sources d'atténuation, l'atténuation due à la viscosité(α_v), celle due à la conductivité thermique(α_{th}) et celle due à la relaxation thermique moléculaire, cette dernière est souvent négligée. Ce coefficient est normalisé par rapport à la fréquence au carrée de l'onde plane. Ce coefficient est sensible à l'humidité pour une température donnée à une fréquence fixe. L'amplitude due à l'humidité peut varier jusqu'à 60% à des températures ambiantes.

Tableau 2-1 : Atténuation par rapport à la fréquence et l'humidité

fréquence \ humidité	0%	40%	100%
$f_c = 10 \text{ kHz}$	0.03 dB/m	0.15 dB/m	0.06 dB/m
$f_c = 20 \text{ kHz}$	0.08 dB/m	0.48 dB/m	0.21 dB/m
$f_c = 40 \text{ kHz}$	0.26 dB/m	1.10 dB/m	0.83 dB/m

On peut écrire l'équation générale de propagation de la pression (2-11), où A correspond à l'amplitude de l'onde.

$$P(x, t) = Ae^{j(\omega t - kx)} e^{-\alpha_{us}x} \quad (2-11)$$

La relation qui lie la pression acoustique à l'intensité est donnée en 2-12:

$$I(x) = \frac{1}{t} \int_0^t |P(x, t)| \cdot u \cdot dt \quad (2-12)$$

Avec le déplacement $u = P/Z$ et Z l'impédance acoustique, on aura alors

$$I(x) = I(x = 1) e^{-2\alpha_{us}(x-1)} \quad (2-13)$$

$$P(x) = \frac{A(x=1)}{x} e^{-\alpha_{us}(x-1)} \quad (2-14)$$

Ce calcul précédent comprend l'atténuation due à la distance mais pas à la position polaire (distance et orientation). Dans l'article [Ko96], les auteurs prennent en compte la position polaire (2-15)

$$p(x, \theta) = (P_0 / x) e^{-2\theta^2 / \theta_0^2} e^{-\alpha_{us}(x-1)} \quad (2-15)$$

où P_0 et θ_0 sont respectivement des constantes de pression acoustique à 1 mètre et de l'angle de dispersion.

On peut remarquer que la pression est inversement proportionnelle à la distance. Dans le cas où l'angle θ est nul, l'onde acoustique a une atténuation dans l'air en $1/x$ pour un trajet simple et en $1/x^2$ pour un aller retour si la réflexion est totale. Si nous utilisons un amplificateur variable dans le temps pour compenser l'action de la distance, on peut écrire l'atténuation due à l'orientation à l'émission et à la réception, respectivement en (2-16) et (2-17). Dans le cas d'un transducteur qui détecte le même objet, l'angle est le même et on écrit l'équation (2-18).

$$P_a(\theta) = e^{-2\theta^2 / \theta_0^2} \quad (2-16)$$

$$P_r(\theta) = e^{-2\theta^2 / \theta_0^2} \quad (2-17)$$

$$P_{ar}(\theta) = P_a(\theta)P_r(\theta) = e^{-4\theta^2 / \theta_0^2} \quad (2-18)$$

2.3 Modélisation du transducteur

2.3.1 Le transducteur

Il existe sur le marché trois grandes familles de transducteur :

- piézo-électriques (effet piézo-électrique)
- capacitifs (effet électrostatique)
- magnétostrictifs (composants ferromagnétiques)

Pour les mesures de distance sans contact, on retrouve les deux premières familles. La première est plus résistante aux variations atmosphériques (le transducteur est fabriqué en céramique et peut-être clos). Cependant la seconde est légèrement moins chère et utilisée dans les applications intérieures. Dans le chapitre sur les capteurs ultrasonores de [Irwing97], un paragraphe entier fait la comparaison entre les deux premières familles et conclut de la façon suivante : les capteurs électrostatiques présentent une meilleure sensibilité et une plus grande bande passante et les capteurs piézo-électriques possèdent une commande simple compatible CMOS. Les magnétostrictifs sont très peu utilisés en mesure, ils reposent sur une interaction courant électrique/champ magnétique (principe du microphone) et il n'est pas possible de mesurer précisément un temps de vol, on les retrouve pour le nettoyage ultrasonore.

2.3.2 Choisir son transducteur

Chaque type de transducteur possède sa sensibilité à l'émission, à la réception, son angle d'ouverture et l'angle correspondant au passage du premier zéro acoustique. Ces différentes caractéristiques sont données dans les fiches techniques du transducteur. L'influence de l'orientation (θ) sur le transducteur peut être écrite sous forme de pression en (2-19).

$$P(\theta) = \frac{2J(k.a.\sin \theta)}{k.a.\sin \theta} \quad (2-19)$$

où $J()$ est la fonction de Bessel de radiation, k est le nombre d'ondes : $k = 2\pi / \lambda = 2\pi * f_e / C_a$, avec la longueur d'onde λ , la fréquence d'émission f_e , la célérité du son C_a et le rayon du transducteur a .

On résout l'équation en utilisant les fonctions étendues de Bessel pour trois termes. Le résultat de la fonction de Bessel pour le transducteur Massa TR89 est 1,62. On trouve le demi-angle d'ouverture θ_0 avec l'équation (2-20). La caractéristique donnée en la figure 2-9 correspond à un transducteur Polaroid 600 pour une atténuation normalisée et une fréquence de 50kHz.

$$\theta_0 = \sin^{-1}(1.62 / k.a) \quad (2-20)$$

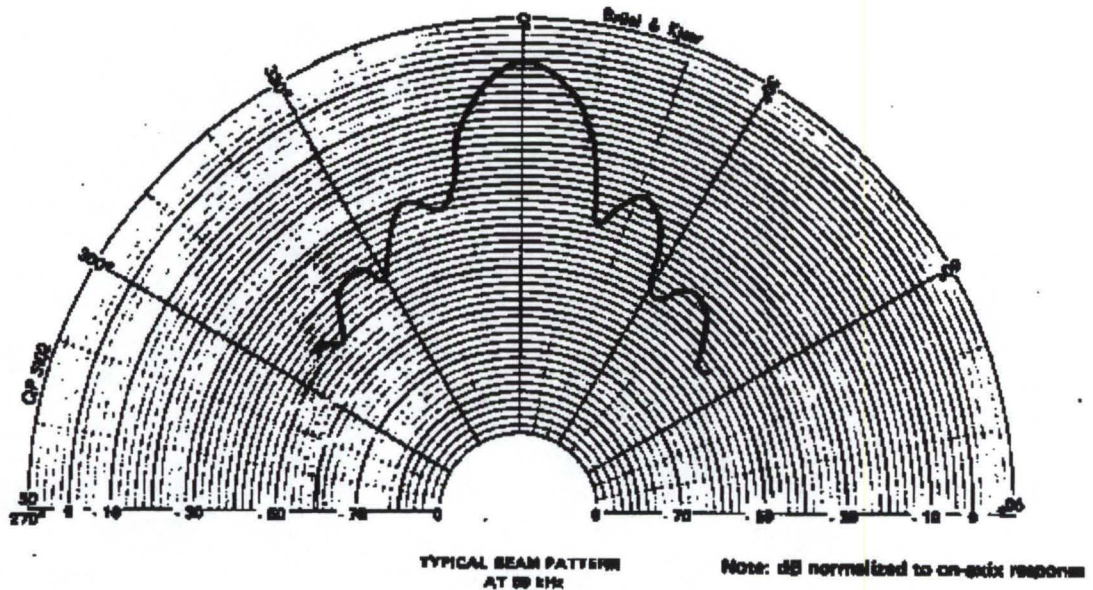


Figure 2-9 : Atténuation de l'onde pour le transducteur Polaroid[Polaroid83]

Remarque : La précision du type du transducteur est importante pour la formule, car chaque transducteur a sa propre fréquence et son diamètre.

Une autre caractéristique importante est la distance L_0 correspondant à la limite entre les champs acoustiques proche et lointain. Le champ proche ou zone de Fresnel est caractérisé par des amplitudes irrégulières. Dans cette zone il est impossible d'avoir une mesure de distance. Le champ lointain ou zone de Fraunhofer est caractérisé par une amplitude des ondes qui décroît de façon monotone suivant l'axe du transducteur et proportionnellement à la distance. La formule (2-21) calculant la distance qui correspond à la limite de la zone de Fresnel, est illustrée par la figure 2-10.

$$L_0 = \frac{(2a)^2}{4\lambda} \quad (2-21)$$

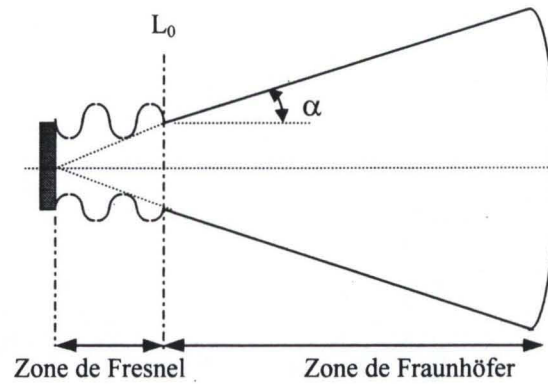


Figure 2-10 : Les champs acoustiques

2.3.3 Modèle physique

2.3.3.1 La relation inverse

La relation inverse, figure 2-11, correspond à l'émission d'ondes ultrasonores. On excite le transducteur à l'aide d'un signal électrique et celui-là vibre, il s'agit d'une déformation u (2-22), pour créer une force (2-23 et 2-24).

$$u = d_r \cdot V \quad (2-22)$$

$$F = k_r \cdot u \quad (2-23)$$

$$F = d_r \cdot k_r \cdot V \quad (2-24)$$

Avec k_r le coefficient de rigidité du cristal (du transducteur) et d_r le rapport de sensibilité du

transducteur : $d_r = \frac{K_r}{k_r}$.

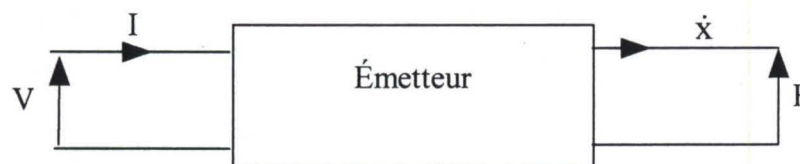


Figure 2-11 : Schéma bloc du transmetteur [Bentley95]

2.3.3.2 La relation directe

La relation directe, figure 2-12, correspond à la réception des ondes ultrasonores. La force acoustique excite le transducteur qui vibre, ce qui provoque la déformation u et la relation directe transforme cette vibration en un signal électrique dont q est la charge électrique (2-24).

$$q = d_r \cdot F \quad (2-25)$$

En utilisant l'équation qui relie la force F à la déformation u et une relation fondamentale de l'électricité $I(t) = \frac{d}{dt} q(t)$ on trouve (2-26) :

$$I = d_r \cdot k_r \cdot \dot{x} \quad (2-26)$$

où $u = \dot{x}$ (vitesse de la déformation)

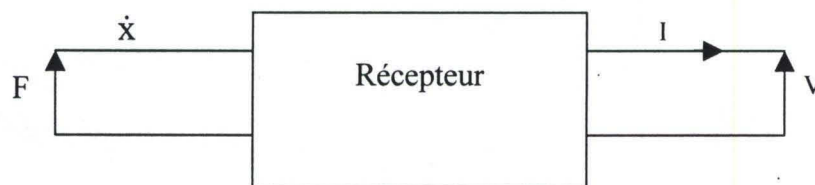


Figure 2-12 : Schéma bloc du récepteur [Bentley95]

Le modèle mécanique d'un transducteur, figure 2-13, est un système masse ressort avec amortisseur (2-27).

$$m\ddot{x}(t) + \lambda_m \dot{x}(t) + k_m x(t) = F(t) \quad (2-27)$$

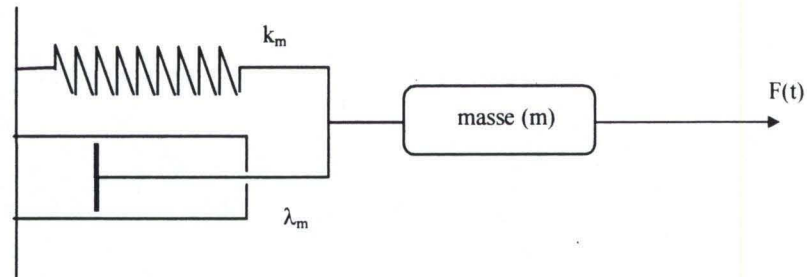


Figure 2-13 : Modèle mécanique du transducteur [Frey69]

où λ_m est la résistance mécanique du système (kg/s).

k_m est la raideur.

2.3.4 Modèle électrique

Tableau 2-2 : Equivalence des grandeurs acoustiques, mécaniques, électriques [Frey69]

Mécanique	Acoustique	Électrique
Masse	inertance	inductance
élasticité	élasticité acoustique	capacitance
résistance	résistance	résistance

Le modèle électrique de la structure mécanique est illustré sur la figure 2-14.

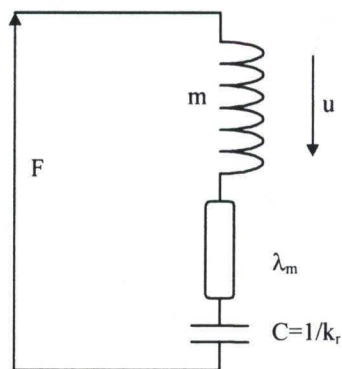


Figure 2-14 : Modèle électrique de la structure mécanique [Frey69]

Les schémas électriques ainsi que les fonctions de transfert sont donnés en annexe A.

2.4 Techniques de mesure de la distance

2.4.1 Mesure par le temps de vol

Elle consiste à compter le temps entre le début de l'émission et celui de la réception. Le temps mesuré t correspond au temps de parcours de l'onde, s'il s'agit d'un transducteur, l'onde a effectué deux fois la distance qui la sépare de l'objet détecté. Son trajet a été effectué à une vitesse constante, il s'agit de la célérité du son, C_a . La distance parcourue, d , est trouvée en (2-28).

$$d = 0.5 * C_a * t \quad (2-28)$$

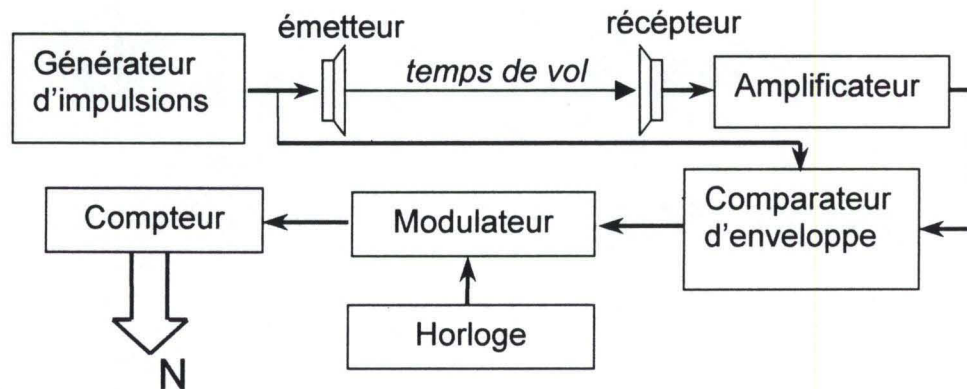


Figure 2-15 : Principe de la mesure par temps de vol

Le temps de vol, pour la figure 2-15 d'après [Kocis96], sera $T_{of} = (N \pm 1) * T_e$ avec T_e la période des impulsions modulées. Pour connaître la distance parcourue par l'onde, il suffit de multiplier le temps trouvé par la vitesse de propagation de l'onde (célérité).

Dans [Irwing97], L. Kleeman classifie trois méthodes pour mesurer le temps de vol en fonction de la taille du traitement (ordre croissant).

(i) *Détection par seuil*

Lorsque le seuil à une amplitude fixe est atteint, le compteur est arrêté, la valeur du compteur correspond au temps de vol de l'onde [Polaroid83]. Cependant, on retrouve plusieurs sources d'incertitude :

- dépendance de l'amplitude en fonction du temps, on corrige cette incertitude par une compensation de l'atténuation avec un gain variable ;
- nature de la surface (coin, plan, ...) fait varier la forme de l'écho;
- dépendance de l'amplitude par rapport à l'angle d'ouverture ;
- niveau de bruit peut altérer la détection.

(ii) *Détection par ajustement de l'enveloppe*

Des techniques d'échantillonnage et un filtre passe bas sont utilisés pour trouver l'enveloppe qui sera normalisée. Le temps d'arrivée de l'écho correspond à la pente la plus élevée de l'enveloppe.

(iii) *Détection par filtrage adapté*

Le filtrage adapté, en anglais *matched filtering*, recherche la corrélation entre un écho reçu (avec bruit) et une forme d'écho typique (sans bruit). Le temps d'arrivée correspond à la position décalée par le temps de l'impulsion typique qui donne le maximum du signal d'intercorrélation [Parrilla91]. Cette technique est très utilisée pour les radars mais la mesure est légèrement altérée par un bruit blanc.

2.4.2 *Mesure par la différence de phase*

Elle consiste à mesurer la phase entre le signal émis et celui reçu. On rappelle que dans la mesure ultrasonore, la fréquence du signal modulé ne change pas avec la réflexion et la propagation. Ainsi avec la mesure de nombreux points (plusieurs par période), on pourra

avoir une information correcte sur la phase.

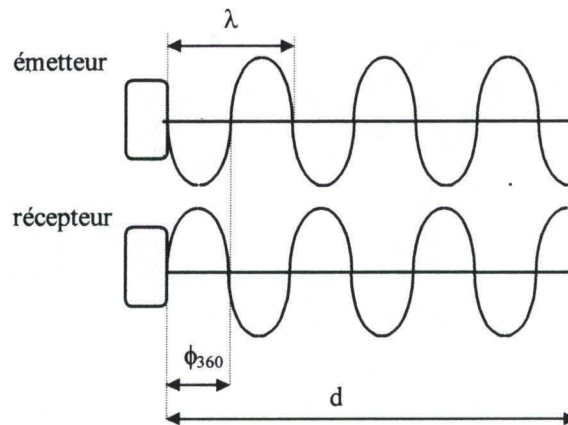


Figure 2-16 : Méthode de différence de phase

La distance d illustrée en 2-16 sera évaluée à l'aide de (2-29) et (2-30) [Everett95] :

$$\cos \phi = \cos\left(\frac{4\pi \cdot d}{\lambda}\right) = \cos\left(\frac{2\pi(\phi_{360} + N\lambda)}{\lambda}\right) \quad (2-29)$$

$$d = \frac{\phi \lambda}{4\pi} = \frac{\phi C_a}{4\pi f_e} \quad (2-30)$$

avec ϕ_{360} : la différence de phase, modulo 360 degrés

N : le nombre de cycles complétés

ϕ : la différence de phase

λ : la longueur d'onde de la modulation

f_e : la fréquence de la modulation

C_a : la célérité du son

Pour connaître la distance supérieure à la longueur d'onde à partir de ce principe, il faut compter le nombre de cycles entiers puis ajouter la différence de phase, ce qui revient à faire la méthode du temps de vol et celle de la différence de phase.

Une méthode économique en point d'acquisition et en nombre de calcul est proposée par [Gueuning96] grâce à un choix judicieux de la fréquence d'acquisition à $\frac{3}{4}$ de la fréquence d'émission et avec l'utilisation des coefficients de la transformée de Fourier discrète pour évaluer la phase.

2.5 Composition d'un système de mesure ultrasonore

Cette décomposition d'un système de mesure ultrasonore est basée sur un montage commercial : "*designer kit*" de la compagnie Polaroid. Le *designer kit* comprend une petite carte de mesure qui commande le transducteur et détecte l'enveloppe du signal reçu. Cette carte est appelée kit Polaroid; la documentation technique se trouve en annexe G. La deuxième carte effectue la gestion du kit Polaroid et affiche le résultat.

2.5.1 Principe de fonctionnement du système Polaroid

Un signal qui est un train d'impulsions à durée fixe et brève, excite le transducteur qui émet des ondes acoustiques à la même fréquence que le signal excitant. Le récepteur les reçoit et les convertit en tension. Pour calculer la distance parcourue par les ondes, il faudra insérer un amplificateur sélectif et variable sur la fréquence d'émission. Cet amplificateur permet d'obtenir une amplitude stable quelque soit la distance. A l'aide d'un filtrage, l'enveloppe du signal est obtenue. Un compteur s'incrémente dès le début de l'émission jusqu'à ce que le comparateur détecte le signal reçu. La valeur du compteur correspond à la distance parcourue par l'onde acoustique à la vitesse du son (340 m/s).

2.5.2 Décomposition sous forme d'un système de mesure

A partir du principe de fonctionnement et des figures 2-1 et 2-2, les parties faisant la conversion analogique numérique ($A_1, A_2, \dots, A_n / N$) sont identifiées :

- conversion du signal électrique excitant en onde acoustique (émetteur);
- conversion de l'onde acoustique par l'onde acoustique réfléchie (réflexion);

- conversion de l'onde acoustique réfléchiée en signal électrique (récepteur);
- mise en forme du signal électrique (amplificateur);
- lissage du signal électrique (enveloppe);
- comparaison du signal (tension de référence).

De même, on peut identifier la partie reconstitution (N / N) :

- comptage du temps écoulé;
- calcul de la distance à partir du temps mesuré et de la vitesse du son;
- circuit de mise en forme des données.

2.5.3 Décomposition en blocs élémentaires d'un système de mesure

On peut détailler ainsi chaque bloc du système de mesure. Les affirmations suivies du symbole (*) indiquent que cette partie n'est pas incluse dans le circuit conditionneur de Polaroid.

Estimation tient compte des paramètres qui influent sur le résultat, ils sont de deux ordres : (i) environnementaux; la vitesse du son de l'onde ultrasonore est une fonction de la composition de l'air(*), de l'humidité(*) et de la température; (ii) électriques; si l'alimentation faiblit, l'étendue de mesure diminuera (*).

Conversion effectue une série de conversions du signal reçu : une onde acoustique en un signal numérique de mesure brute, ce signal passant au niveau logique 1 lorsqu'une onde est détectée.

On peut donner la série de conversions:

A_1 / A_2 : onde acoustique en signal temporel (vu au paragraphe 2.2.3)

A_2 / A_3 : conversion tension / tension dont l'amplitude du signal reçu est constante dans l'étendue de mesure

A_3 / A_4 : filtrage, on obtient une enveloppe du signal

A_4 / N : tension / logique

Conversion tension / tension à amplitude constante (A_2/A_3)

Connaissant les caractéristiques de la propagation des ondes acoustiques, le gain variable compense en distance l'atténuation. Ce dernier est fait à partir d'un gain variant par palier (tableau 2-3). Le filtre sélectif est un filtre RLC dont le facteur de qualité est élevé. Dans le kit, le gain variable est réalisé par deux circuits intégrés de chez Texas Instruments, le SN28784 (analogique) et le TL851 (digital) qui génère les combinaisons binaires de la fonction escalier.

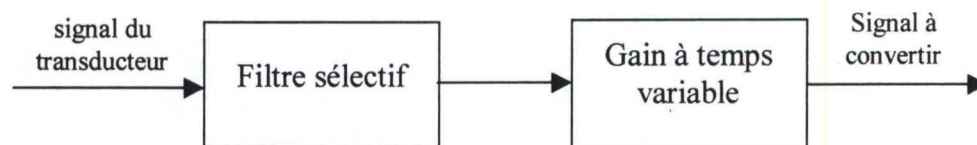


Figure 2-17 : Bloc conversion A_2/A_3

Tableau 2-3 : Correspondance entre l'intervalle temporel et le gain de l'amplificateur

Temps	gain	temps	Gain
0 à 2.38 ms	0	18.84 à 21.58 ms	5
2.38 à 5.12 ms	0.25	21.58 à 27.07 ms	6
5.12 à 7.87 ms	0.5	27.07 à 32.55 ms	7
7.87 à 10.61 ms	1	32.55 à 38.04 ms	8.5
10.61 à 13.35 ms	1.5	38.04 à 43.53 ms	10
13.65 à 16.09 ms	3	43.53 à environ 60 ms	11.75
16.09 à 18.84 ms	4		

Extraction de l'enveloppe du signal (A_3/A_4)

Ce bloc effectue une détection d'enveloppe, puisque le signal envoyé est modulé en amplitude, cette démodulation est aussi faite par SN28784N.

Conversion en numérique (A_4 / N)

La fonction de ce bloc est de comparer l'enveloppe du signal avec un niveau de référence et d'envoyer un signal lorsque le comparateur bascule pour la première fois, cette fonction est réalisée aussi par le circuit TL851.

Stimulation ou excitation (N/A) est le bloc qui émet l'onde ultrasonore, attention, ce dernier est commandé par le bloc contrôle afin que le transducteur ne soit pas en position émission et réception en même temps.

Reconstitution (N / N) est le bloc qui convertit le signal digital en une valeur quantifiée du temps, N divisions unitaires. Cette valeur N, sera ensuite transformée en une valeur de distance, $d = 0.5 * k_c * N * C_a$, où k_c un facteur de conversion qui lie une division unitaire au temps, C_a est la vitesse du son. Le résultat dans le Polaroid série 6500 est codé sur trois digits, qui commanderont trois afficheurs 7 segments. Dans le système de mesure, on peut diviser la reconstitution en deux parties. (i) l'étalonnage consiste à identifier la fonction de transfert d'un bloc si on connaît l'entrée (échantillon de référence) et la sortie; (ii) la reconstitution(*) corrige les données réelles pour qu'elles correspondent à la réalité (échantillon de référence), à partir de la connaissance de la sortie et de la fonction de transfert.

La vitesse du son à température ambiante (20°) est de 343.54 m/s, il faut 53,01 ms pour attendre l'écho le plus loin détecté par le système (30 pieds¹). Le système a une résolution de 1/10 pied, il faudra alors un compteur qui compte 300 coups dans les 53,01 ms, c'est à dire à une fréquence d'horloge de 5,65 kHz.

¹ 1 pied correspond à 12 pouces soit 0,3048 m

Contrôle est un bloc qui a pour fonction de commander l'excitation afin que le transducteur soit en mode émission ou en mode réception.

Ce bloc comprend aussi un circuit de mise en forme qui est destiné au bloc d'excitation, on trouve un compteur qui détermine la fonction porte de durée de l'émission de l'onde, et un oscillateur à quartz qui module le signal porte en amplitude. La fréquence de modulation $f_c = 49.41 \text{ kHz}$ et sera utilisée pour la réception, afin de filtrer les signaux parasites. Cependant le transducteur électrostatique requiert, pour un fonctionnement normal, une intensité impulsionnelle de 3 Ampères qui est transformée à l'aide d'un transformateur en une tension crête à crête de 300 Volt. De plus, il assure une interdiction d'émission pendant 5 ms à la mise sous tension, à cause des oscillations possibles au niveau du transducteur et de la stabilité des circuits.

Chapitre 3 :

Algorithme de reconstitution de la distance et de fusion

Dans le précédent chapitre, j'ai introduit les algorithmes qui sont nécessaires au système de mesure. Le premier permet de mesurer le temps de vol (la distance, paragraphe: "2.4.1 mesure par temps de vol"). La mesure de niveau d'enveloppe pour déclencher le temps de vol manque de précision, une méthode plus robuste est la méthode de l'intercorrélation pour définir le temps de vol. Le deuxième algorithme a pour but d'augmenter la précision de la localisation des objets détectés. Le nombre d'éléments sensoriels requis est proportionnel au nombre de degrés pour la localisation (1 transducteur: mesure de distance; 2 transducteurs: mesure de position dans un repère 2 dimensions...). On utilisera un algorithme pour déterminer l'équation de la tangente de l'objet détecté.

3.1 Reconstitution de la distance

3.1.1 Principe

Dans le chapitre précédent, les méthodes de temps de vol ont été classifiées en trois catégories. La dernière catégorie, détection par filtrage adapté, détermine de façon plus précise le temps de vol. Cette dernière est effectuée à l'aide de l'intercorrélation entre le signal émis et celui reçu. Une réalisation expérimentale de cette technique est décrite dans [Gueuning96].

3.1.1.1 L'intercorrélation

L'intercorrélation donne une information sur la ressemblance entre deux fonctions, cette ressemblance se concrétisant par un ou des maximums. L'intercorrélation entre les vecteurs numériques g et h , respectivement de dimension n et o , est notée C_{gh} .

$$C_{gh}(j) = \sum_{i=0}^{n-1} g(i) * h(i+j) \quad (3-1)$$

La dimension de C_{gh} est de $2*max-1$ avec max la dimension la plus grande entre n et o . L'échantillon max correspond à un décalage de 0 et le premier échantillon correspond à un décalage de $-max+1$.

L'application traitée est causale où le signal arrive toujours après le signal émis. Alors, il est possible de calculer l'intercorrélation sur un nombre max de points (max à $2*max-1$), c'est à dire d'un retard nul à un retard de $max-1$.

Dans la même optique de réduction du nombre de points à mémoriser, on doit faire l'intercorrélation uniquement avec les points représentant le signal reçu, les autres points sont mis à zéro et représentent environ 98% du temps de vol limite, en réalité pour ces points, on mesure le bruit de mesure. Pour obtenir le résultat de l'intercorrélation, on effectue le calcul avec l'information utile puis on ajoute un nombre d'échantillons à zéro représentant le délai entre le signal émis et reçu.

Si le signal émis a la même forme que le signal reçu, le résultat d'intercorrélation donnera exactement le délai en échantillons. Nous touchons, ici, au point problématique, car la forme du signal reçu dépend de la forme de l'objet et de sa position (orientation) et du transducteur avec l'électronique associée.

Dans ce chapitre, nous allons expliquer le prétraitement effectué sur les deux variables, pour calculer l'intercorrélation.

3.1.1.2 Méthode de calcul de l'intercorrélation en ligne/hors ligne

Le calcul de l'intercorrélation peut être fait au fur et à mesure que les échantillons sont reçus, on appelle ce type de méthode en ligne. Avec la méthode hors ligne, le calcul de l'intercorrélation est effectué après le dernier échantillon reçu. Cette dernière se calcule à partir de la définition mathématique de l'intercorrélation à l'aide de deux boucles imbriquées.

Pour j allant de 1 par pas de 1 jusqu'à o
 Pour i allant de 1 par pas de 1 jusqu'à n
 $C_{gh}(j) = C_{gh}(j) + g(i) h(i+j-1)$
 fin
fin

Cet algorithme peut se décliner en une version en ligne

Pour j allant de 1 par pas de 1 jusqu'à o
 Pour i allant de 1 par pas de 1 jusqu'à n
 $C_{gh}(j+i-1) = C_{gh}(j+i-1) + g(n+1-i) * h(j)$
 fin
fin

Le choix de la méthode (hors/en ligne) dépend de la rapidité de la structure matérielle, le nombre d'opérations pour la méthode en ligne est plus important (plus d'accès mémoire). Le choix est effectué dans le chapitre suivant.

3.1.1.3 Le prétraitement avant l'intercorrélation

Ce prétraitement consiste premièrement par l'échantillonnage de nos signaux. L'échantillonnage est régi par le théorème de Nyquist qui demande que la fréquence d'échantillonnage soit au moins deux fois plus importante que celle du signal, pour pouvoir retrouver le signal. Dans l'article [Gueuning96], l'auteur propose d'effectuer l'échantillonnage tous les $3/4$ de période. Le théorème n'est pas respecté, cet échantillonnage fournit des données pour l'intercorrélation non pas pour la forme du signal. Pour cette raison, la contrainte sur l'échantillonnage est caduque. La fréquence d'acquisition choisie dans l'article est justifiée uniquement pour limiter le nombre de calculs dans la méthode de différence de phase. Cette dernière n'est pas retenue dans le cadre du mémoire. On étudiera l'influence de la période d'échantillonnage sur le temps de vol trouvé.

Cependant, on peut essayer diverses opérations mathématiques sur les signaux échantillonnés. Dans l'article de [Gueuning96], le signal reçu est mis au carré et l'on garde uniquement la fonction porte du signal émis. De nombreuses autres possibilités existent, elles sont simulées dans le paragraphe suivant.

3.1.2 Simulation

A l'aide du logiciel Matlab, des schémas équivalents donnés par [Bentley95] ainsi que les spécifications techniques du transducteur électrostatique série 600 [Polaroid82], un modèle synthétique du transducteur (annexe A) est défini de façon à ce que la réponse ressemble à l'écho acquisitionné (figure 3-1). Ce modèle permet de tester différentes solutions afin de trouver les meilleurs facteurs (nombre de cycles émis, période d'échantillonnage, fonctions de prétraitement, influence du second écho) qui donnent la bonne évaluation du temps de vol. L'ensemble des programmes ainsi que des informations complémentaires sont disponibles en annexe B.

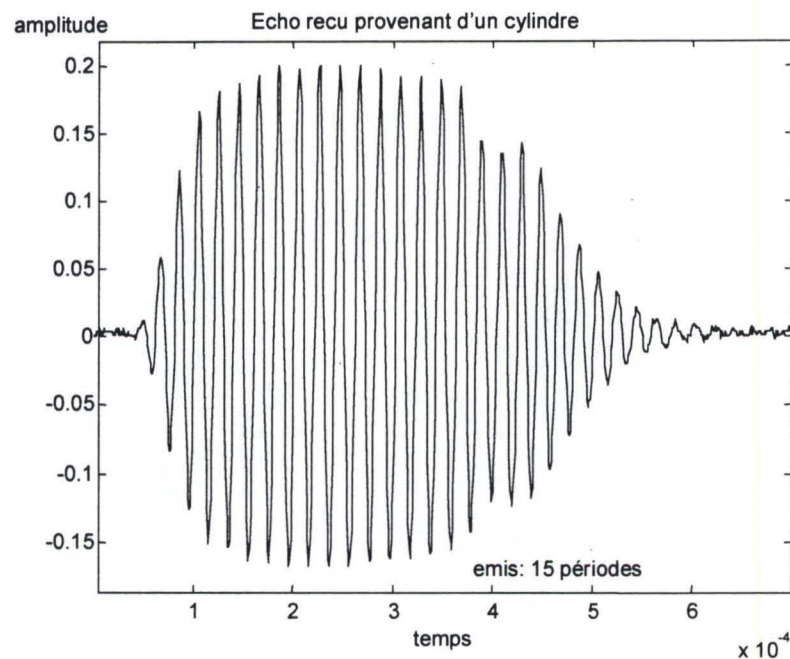


Figure 3-1 : Écho mesuré provenant d'un cylindre

3.1.2.1 Fonctions de prétraitement

Ce premier facteur est déterminant pour obtenir un bon temps de vol. Une série de combinaisons est présentée. Pour chaque résultat, différentes phases allant par pas de $1/48$ de période sont ajoutées au signal reçu car l'acquisition de l'écho ne commence pas obligatoirement par une phase nulle. Les colonnes reçues et émises indiquent l'opération effectuée sur le signal. Pour les tableaux 3-1 et 3-2, le signal émis est respectivement un sinus échantillonné à $4/3$ de la fréquence d'émission et la fonction porte de largeur l'émission qui est ici de 15 impulsions. La valeur de la mesure exacte à trouver est 21 échantillons. Le chiffre 1 indique que le signal ne subit aucun traitement, les codes "carré" et "abs" correspondent respectivement à la mise au carré du signal et au calcul de la valeur absolue. Les autres colonnes contiennent des statistiques sur les résultats de simulation soit respectivement la moyenne, la valeur maximale, la valeur minimale et l'écart-type (échantillon de mesure).

Tableau 3-1 : Résultats de la méthode avec sinus échantillonné à $3/4$ de T

reçu	émis	moyenne	max.	min.	écart type
carré	carré	24	25	23	0.7928
carré	1	26.4444	31	11	7.5874
abs.	abs.	24	25	23	0,8261
abs.	1	21,3889	30	11	9,0468
1	1	24.0556	25	23	0.7908

Tableau 3-2 : Résultats de la méthode avec fonction porte

reçu	émis	moyenne	max.	min.	écart type
carré	1	24.3889	26	24	0.5492
abs.	1	24.3889	26	24	0.5492
1	1	27,6666	31	11	9.6385

Remarque: Pour toutes les simulations, le décalage entre la valeur à trouver et celle calculée est constante (i.e. si on ajoute n retards, le résultat aura n retards de plus que le résultat calculé pour un retard de 0). Il suffit dans le traitement du résultat d'ajouter ce délai.

Le critère de choix des fonctions de prétraitement est fait de façon que les résultats aient une faible dispersion. Le premier tableau donne des résultats corrects, excepté là où le prétraitement est différent entre émis et reçu. Le deuxième tableau donne des résultats très bons avec un prétraitement qui effectue la mise au carrée ou le calcul de la valeur absolue. Sans prétraitement le résultat n'est pas exploitable. Les résultats obtenus avec la mise au carré ou le calcul de la valeur absolue sont identiques quels que soient le retard et la forme. On peut effectuer une interpolation parabolique autour du maximum (3 points), mais les résultats obtenus sont légèrement inférieurs car l'écart type augmente un peu pour une moyenne qui ne change quasiment pas. De plus, cette interpolation demande des instructions supplémentaires.

La combinaison choisie est l'intercorrélation entre la fonction porte du signal émis et la valeur absolue du signal reçu.

3.1.2.2 Influence du nombre de cycles émis

Le deuxième test consiste à évaluer le nombre de cycles à l'émission de façon à obtenir un écho contenant le maximum d'informations. À partir du modèle synthétique, on évalue la position donnée par l'intercorrélation avec les fonctions de prétraitement choisies. Les résultats des simulations sont quasiment identiques pour un nombre de périodes supérieur ou égal à 2.

3.1.2.3 Influence de la fréquence d'acquisition

La fréquence d'acquisition a comme effet de déterminer la résolution de la mesure du temps de vol. Plus le rapport entre la fréquence d'acquisition et celle d'émission est grande, meilleure est la résolution, mais les performances matérielles doivent être importantes. Le but de ce mémoire est de faire un système précis et économique, on limite la fréquence d'acquisition à deux fois la fréquence émise, environ 100 kHz. Ce rapport doit, quelle que soit la phase du système, donner des valeurs représentatives de l'écho. La figure 3-2 donne les valeurs moyennes maximums et minimums pour 15 périodes acquisitionnées avec des phases quelconques en fonction du rapport entre la période d'acquisition et celle d'émission.

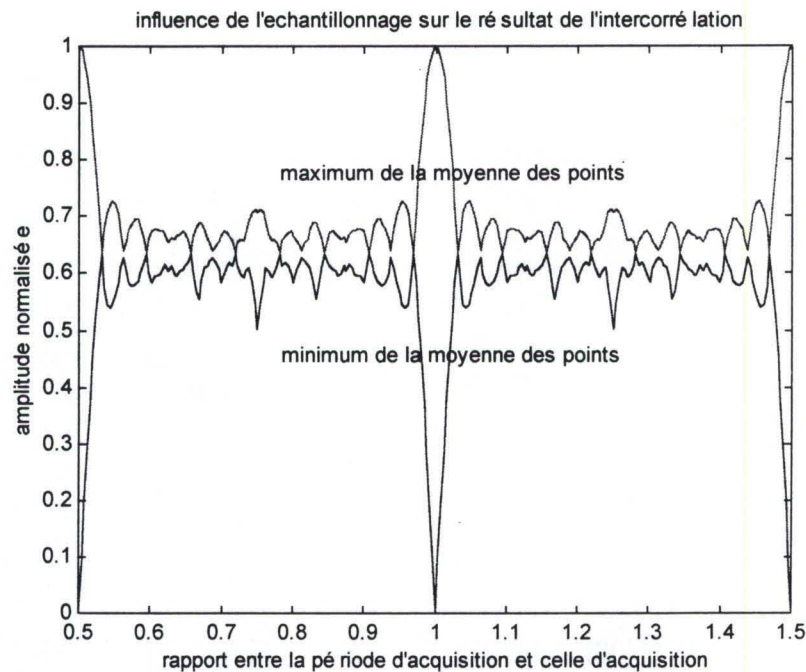


Figure 3-2 : Influence de l'échantillonnage sur le résultat de l'intercorrélation

On remarque que les multiples de 0.5 donnent des mauvais résultats si la phase est égale à un multiple de π , les points sont nuls. De même les résultats proches de ces multiples et des valeurs de $2/3$, $3/4$ et $5/6$ donnent une dispersion plus importante que les autres rapports.

3.1.2.4 Influence d'un deuxième écho proche

Le but de ce test est d'évaluer l'influence d'un deuxième écho sur la position de l'écho donné. On utilisera différentes valeurs de retard et des multiples de l'amplitude du premier écho pour le deuxième écho.

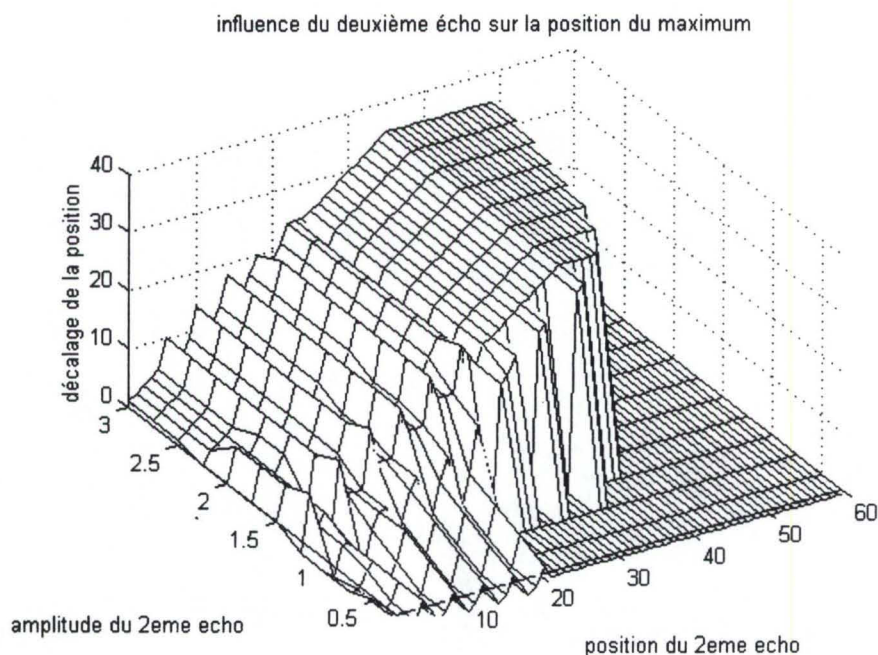


Figure 3-3 : Influence du 2eme echo sur l'intercorrélation

Ces tests ont été effectués avec une période d'acquisition de $\frac{3}{4}$ de la période émise et d'une longueur de l'acquisition de 2 fois le signal émis échantillonné. Ce dernier paramètre est important uniquement pour le cas d'un deuxième écho. Dans le cas d'un simple écho, aucune influence sur la position du résultat est remarquée pour des longueurs de 2 fois et plus, seule la mémoire et la quantité de calcul augmentent. L'interprétation de la figure 3-3 indique que le positionnement avec l'intercorrélation est toujours correct s'il n'y a pas de deuxième écho dans la largeur de la fenêtre d'acquisition. Dans le cas contraire, le résultat n'est pas garanti, il faudra dissocier les deux échos et les traiter séparément.

En conclusion, l'intercorrélation avec les facteurs spécifiés présente une grande robustesse pour positionner le début d'un écho reçu.

3.1.3 Algorithme de mesure de distance

On remarque dans la description du fonctionnement du kit de Polaroid, qu'il n'y a pas

toujours un séquentiellement entre les fonctions. Par exemple, le transducteur attend une réception et le compteur compte le temps écoulé. En réalité il y a aura plusieurs programmes qui fonctionnent en parallèle. Il est plus simple de représenter cet algorithme sous forme d'un organigramme, tel qu'illustré à la figure 3-4.

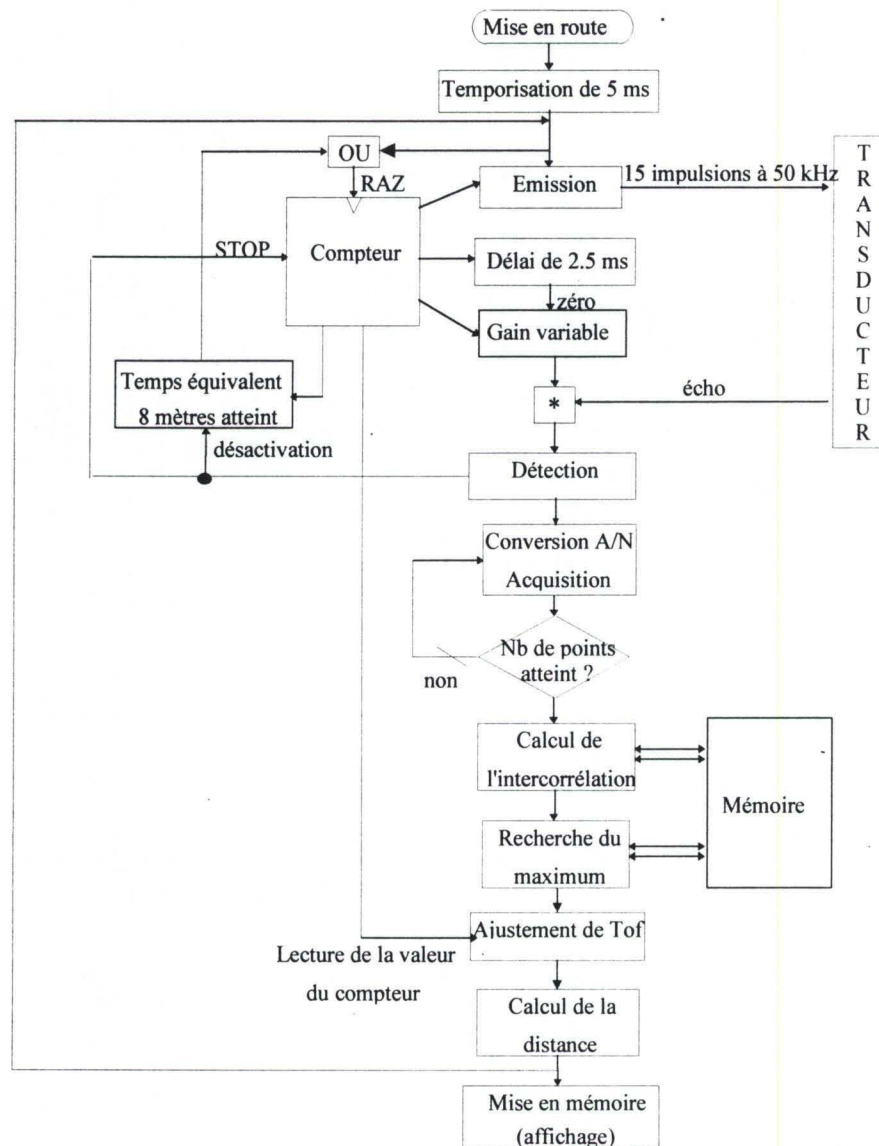


Figure 3-4 : Organigramme du programme pour la mesure de distance

3.2 Algorithme de fusion

3.2.1 Principe

L'algorithme de fusion permet d'estimer la tangente de l'objet détecté par deux capteurs ultrasonores à l'aide d'un filtre de Kalman étendu, il s'agit d'une version du filtre de Kalman qui est linéarisé.

L'équation de la tangente d'ordre 1 correspond à l'équation du segment de droite détecté $l = p.x_M + q.y_M$, où les coordonnées x_M et y_M sont déterminées par géométrie en fonction de la distance donnée par les deux transducteurs (d_1 , d_2) et par l'orientation des transducteurs (θ) et de leur position respective dans le repère.

Le système sensoriel est disposé comme illustré à la figure 3-5.

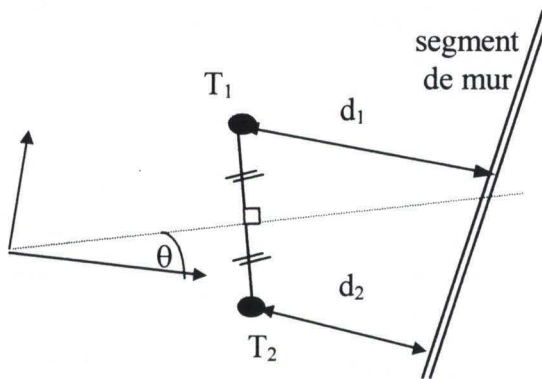


Figure 3-5 : Représentation du système sensoriel de mesure de position

3.2.1.1 Rappel sur le filtre de Kalman linéaire

Soit un système d'équations d'état à temps discret :

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{w}(k) \quad (3-2)$$

$$y(k+1) = C(k+1)x(k+1) + v(k+1) \quad (3-3)$$

Où $x(k)$ la variable d'état à l'instant k

$A(k)$ la matrice de transition

$B(k)$ le vecteur d'entrée

$C(k)$ le vecteur de mesure

$y(k)$ la sortie du système

$w(k)$ et $v(k)$ respectivement le bruit blanc d'entrée et de mesure qui sont statistiquement indépendants

Les équations du filtre de Kalman qui estiment ce système, sont données par :

$$\hat{x}(k) = \hat{x}(k-1) + K(k)[y(k) - C(k)\hat{x}(k-1)] \quad (3-4)$$

$$K(k) = P(k-1)C^T(k)[C(k)P(k-1)C^T(k) + R(k)]^{-1} \quad (3-5)$$

$$P(k+1/k+1) = [I - K(k+1)C(k)] * P(k+1/k) \quad (3-6)$$

$$P(k+1/k) = A(k)^T P(k) A(k) + B(k)^T Q(k) B(k) \quad (3-7)$$

Ces équations sont les mêmes que celle de [Najim88], à part les notations abrégées tel que $\hat{x}(k+1)$ correspond dans le livre à $\hat{x}(k+1/k)$

La matrice $R(k) = E[v(k)v(k)^T]$ est la matrice autocorrélation du bruit de mesure

La matrice $Q(k) = E[w(k)w(k)^T]$ est la matrice autocorrélation du bruit d'entrée

La matrice $P(k) = E[x(k/k)x(k/k)^T]$ est la matrice de covariance de l'erreur d'estimation

La condition nécessaire pour assurer la convergence du filtre est l'observabilité du système d'état dans le cas de la reconstitution d'état (dans cet algorithme, nous allons estimer les coefficients de la tangente qui sont nos variables d'état).

3.2.1.2 Modèle non linéaire

Le filtre de Kalman pour les systèmes non linéaires repose sur une application du filtre de Kalman linéaire sur un modèle linéarisé à partir de la mesure actuelle et des valeurs estimées à l'état précédent.

3.2.1.3 Algorithme d'estimation des coefficients

Pour chaque nouvelle mesure, on effectue une linéarisation de la fonction non linéaire à l'aide du vecteur de mesures actuelles et du dernier vecteur des estimés. Dans le cas d'un nombre d'itérations supérieure à 1, on recalcule les estimés de nouveau à partir des estimés précédents. Cet algorithme est écrit sous forme d'organigramme en la figure 3-6.

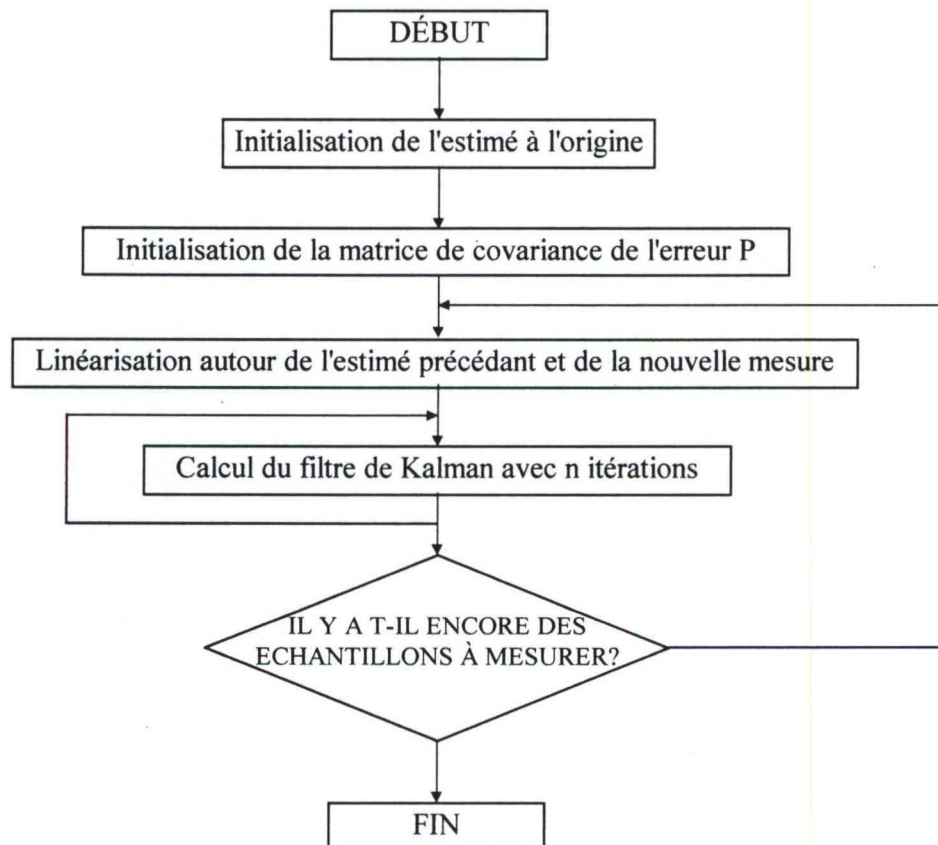


Figure 3-6 : Organigramme de l'algorithme d'estimation des paramètres de la droite

3.2.2 Algorithme de Kalman étendu pour la perception d'objet

Cette section est adaptée de l'article de Song et Tang publié dans la revue *IEEE Transactions on industrial electronics* de juin 1996 [Song96].

3.2.2.1 Étude de la géométrie

Le principe de base de la mesure de distance utilisant un transducteur ultrasonore, est la mesure de la différence de temps entre l'émission et la réception de l'écho, car la vitesse de propagation acoustique est constante. Cependant, les ondes acoustiques ne se propagent pas dans une même direction, mais se dispersent un peu. On notera β l'angle de dispersion des ondes. En tenant compte de cette dispersion, le temps trouvé qui est relatif à la distance la plus courte, n'est pas toujours la distance de l'objet se trouvant sur la normale du transducteur OA , mais la distance la plus courte du transducteur $OA' = OA'$ se trouvant dans la zone mesure, délimitée par l'angle β et par la portée du transducteur.

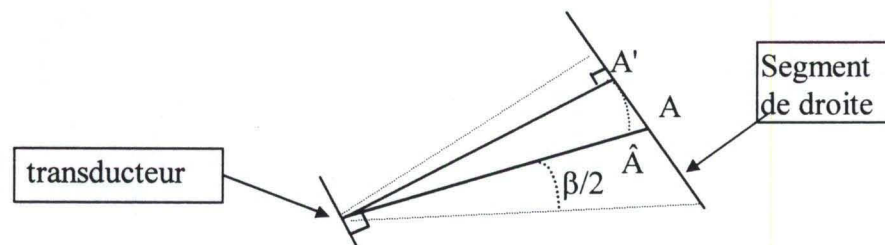


Figure 3-7 : Réflexion sur un mur

Le système sensoriel pour l'estimation de la tangente consiste à placer deux transducteurs ultrasonores sur le même axe et d'estimer la distance entre l'objet et le milieu des deux transducteurs. Ces deux derniers seront éloignés de la distance s_d .

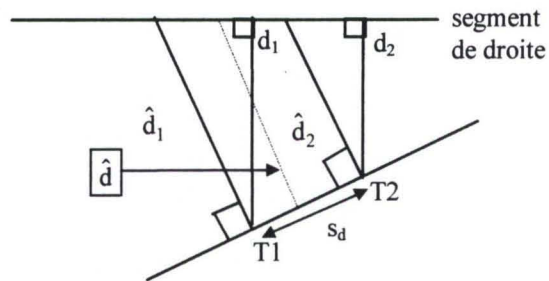


Figure 3-8 : Système de mesure à double transducteur

En fonction des deux distances mesurées et de l'espacement des deux transducteurs, s_d , on définit l'angle Φ qui vérifie si la mesure est comprise dans le demi-angle d'ouverture.

$$\tan(\Phi) = \frac{|d_1 - d_2|}{s_d} \quad (3-8)$$

Il faut prendre $|d_1 - d_2|$ comme une distance et non pas comme une valeur algébrique pour appliquer la relation trigonométrique.

Pour recevoir la réflexion de l'objet, il est nécessaire d'avoir Φ inférieur à θ (angle entre l'axe $[Ox)$ du repère et la normale aux transducteurs). Dans la suite du travail, je pose que la distance minimale détectée par le transducteur correspond à la distance se trouvant perpendiculaire à la surface du mur, c'est à dire $\Phi > \beta/2$.

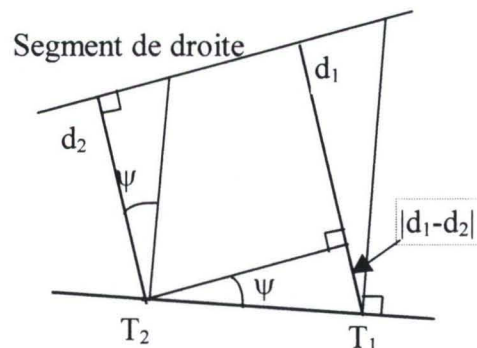


Figure 3-9 : Géométrie du système à double transducteur

ψ représente l'angle d'ouverture entre la direction de l'objet détecté par le transducteur et celle correspondant à la normale du transducteur. ψ est déterminé par la relation trigonométrique (3-9). A partir des figures 3-8 et 3-9, on peut déterminer la distance \hat{d} passant par le milieu des deux transducteurs (3-9).

$$\sin(\psi) = \frac{|d_1 - d_2|}{s_d} \quad (3-9)$$

$$\hat{d} = \frac{(d_1 + d_2)}{2 \cdot \cos \psi} \quad (3-10)$$

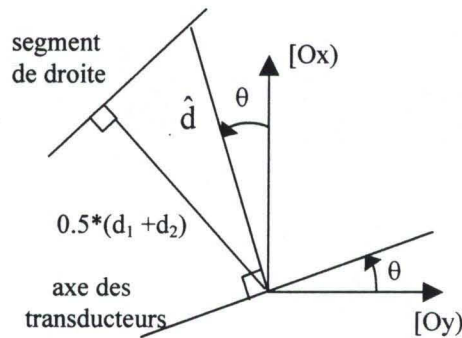


Figure 3-10 : Le repère mobile

Considérons le repère $\{[Ox];[Oy]\}$ avec θ l'angle formé par l'axe des transducteurs et l'axe du repère $[Oy]$, comme sur la figure 3-10. Les coordonnées (x_M, y_M) à l'intersection de \hat{d} et la tangente de l'objet sont données par trigonométrie en (3-11 et 3-12).

$$x_M = \hat{d} \cdot \cos(\theta) \quad (3-11)$$

$$y_M = \hat{d} \cdot \sin(\theta) \quad (3-12)$$

3.2.2.2 Linéarisation du système

Le système non linéaire $f(a,b) = p \cdot x_M + q \cdot y_M - 1 = 0$ est linéarisé autour de (a_0, b_0) à l'aide

d'une expansion en série de Taylor d'ordre 1.

$$\mathbf{f}(\mathbf{a}, \mathbf{b}) \cong \mathbf{f}(\mathbf{a}_0, \mathbf{b}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \right|_{\mathbf{a}_0, \mathbf{b}_0} (\mathbf{a} - \mathbf{a}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{b}} \right|_{\mathbf{a}_0, \mathbf{b}_0} (\mathbf{b} - \mathbf{b}_0) \quad (3-13)$$

Où \mathbf{a}_0 est le vecteur d'état $[\mathbf{p}, \mathbf{q}]^T$ et \mathbf{b}_0 le vecteur des données $[d_1, d_2, \theta]^T$

Le filtre de Kalman linéaire est défini au début du sous chapitre, dans cette application, le vecteur d'état \mathbf{a} est mis à jour avec $\hat{\mathbf{x}}(\mathbf{k}) = \hat{\mathbf{a}}(\mathbf{k})$ ce qui signifie que la matrice d'état \mathbf{A} est une matrice identité. Le système d'état est considéré sans bruit d'entrée mais uniquement avec du bruit de mesure, donc $\mathbf{B}(\mathbf{k}) = 0$.

$$\hat{\mathbf{x}}(\mathbf{k}) = \hat{\mathbf{x}}(\mathbf{k} - 1) + \mathbf{K}(\mathbf{k})[\mathbf{y}(\mathbf{k}) - \mathbf{C}(\mathbf{k})\hat{\mathbf{x}}(\mathbf{k} - 1)] \quad (3-14)$$

En tenant compte des simplifications sur le système d'état, les deux matrices de covariance de l'erreur $\mathbf{P}(\mathbf{k}+1/\mathbf{k})$ et $\mathbf{P}(\mathbf{k}+1/\mathbf{k}+1)$ peuvent être fusionnées en une matrice unique $\mathbf{P}(\mathbf{k})$.

$$\begin{cases} \mathbf{P}(\mathbf{k}+1/\mathbf{k}) = \mathbf{A}(\mathbf{k})^T \mathbf{P}(\mathbf{k}/\mathbf{k}) \mathbf{A}(\mathbf{k}) + \mathbf{B}(\mathbf{k})^T \mathbf{Q}(\mathbf{k}) \mathbf{B}(\mathbf{k}) \end{cases} \quad (3-15)$$

$$\begin{cases} \mathbf{P}(\mathbf{k}+1/\mathbf{k}+1) = [\mathbf{I} - \mathbf{K}(\mathbf{k}+1)\mathbf{C}(\mathbf{k})]^* \mathbf{P}(\mathbf{k}+1/\mathbf{k}) \end{cases} \quad (3-16)$$

$$\mathbf{P}(\mathbf{k}) = [\mathbf{I} - \mathbf{K}(\mathbf{k})\mathbf{C}(\mathbf{k}-1)]\mathbf{P}(\mathbf{k}-1) \quad (3-17)$$

Les autres termes ne changent pas. Le système linéarisé est identifié :

$$\mathbf{y}(\mathbf{k}) = \mathbf{C}(\mathbf{k})\mathbf{x}(\mathbf{k}) + \mathbf{v}(\mathbf{k}) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \right|_{\mathbf{a}_0, \mathbf{b}_0} \mathbf{a} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{b}} \right|_{\mathbf{a}_0, \mathbf{b}_0} (\mathbf{b} - \mathbf{b}_0) \quad (3-18)$$

$$\mathbf{y}(\mathbf{k}) = -\mathbf{f}(\mathbf{a}_0, \mathbf{b}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{a}} \right|_{\mathbf{a}_0, \mathbf{b}_0} \mathbf{a}_0 \quad (3-19)$$

En remplaçant x_M et y_M par les coordonnées du point de mesure sur l'axe du transducteur, dans l'expression de $f(\mathbf{a}, \mathbf{b})$ on trouve:

$$f(\mathbf{a}, \mathbf{b}) = p \frac{d1 + d2}{2 \cos \psi} \cos \theta + q \frac{d1 + d2}{2 \cos \psi} \sin \theta - 1 = 0 \quad (3-20)$$

Pour calculer les termes du filtre de Kalman, il suffit d'effectuer l'identification et de remplacer les termes par ceux de l'équation non linéaire. Cependant, le vecteur $\mathbf{v}(\mathbf{k})$ ne peut pas être calculé car le vecteur de \mathbf{b} n'est pas accessible. Dans la résolution du filtre de Kalman, on utilise l'expression de $\mathbf{y}(\mathbf{k})$ en (3-18) au lieu de celle en (3-19). Pour la matrice de covariance de bruit $\mathbf{R}(\mathbf{k}) = \mathbf{E}[\mathbf{v}(\mathbf{k}) \mathbf{v}(\mathbf{k})^T]$, on utilise l'expression suivante.

$$\mathbf{R}(\mathbf{k}) = \left(\left[\frac{\partial \mathbf{f}}{\partial \mathbf{b}} \right]_{\mathbf{a}_0, \mathbf{b}_0} \cdot \text{Variance}(\mathbf{b})^2 \cdot \left[\frac{\partial \mathbf{f}}{\partial \mathbf{b}} \right]_{\mathbf{a}_0, \mathbf{b}_0}^T \right) \quad (3-21)$$

Les autres termes du filtre se calculent en résolvant les différentes équations partielles, les calculs sont présentés dans le paragraphe 3.2.3.

Les dimensions des éléments de l'algorithme sont les suivants :

- Le vecteur estimé $\mathbf{a}(\mathbf{k})$: [2,1]
- Le vecteur $\mathbf{b}(\mathbf{k})$ de mesure : [3,1]
- La matrice $\mathbf{R}(\mathbf{k})$: [1,1]
- La matrice $\mathbf{C}(\mathbf{k})$: [1,2]
- Le gain de Kalman $\mathbf{K}(\mathbf{k})$: [2,1]
- Le vecteur de mesure $\mathbf{y}(\mathbf{k})$: [1,1]
- La matrice $\mathbf{P}(\mathbf{k})$ = [2,2]

3.2.2.3 Observabilité

Dans la présentation du filtre de Kalman pour la reconstruction d'état, une condition nécessaire pour assurer la convergence des estimés est d'avoir le système d'état qui est observable.

Un système d'état est observable si et seulement si le rang de la matrice d'observabilité est égal à la dimension du système d'état. La matrice d'observabilité est de la forme suivante avec i la dimension du système

$$\mathbf{Obs} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{i-1} \end{bmatrix} \quad (3-22)$$

Pour l'application, la dimension est de 2 et la matrice \mathbf{A} est identitée ce qui donnent un rang du vecteur d'observabilité de un, donc le système est non observable. Des simulations ont été effectuées avec le système et les estimés tendent vers leur première valeur finale et deviennent après, insensibles au changement d'environnement. Des éléments du gain de Kalman et de la matrice de covariance tendent rapidement vers 0. Il est nécessaire de rendre le système observable. Dans le paragraphe suivant une méthode sera proposée.

3.2.2.4 Algorithme de Kalman avec facteur d'oubli

La méthode optée pour rendre observable notre système est une méthode avec facteur d'oubli et de mise à la valeur initiale. Le terme anglais pour cette méthode est *Exponential Forgetting and Resetting Algorithm* ou EFRA. Cette méthode est appliquée à un filtre RLS, *Recursive Least Square* qui est un algorithme des moindres carrés récurrents. Ce dernier est très proche du filtre de Kalman; on retrouve la théorie de cette méthode dans l'article de Salgado, Goodwin et Middleton [Salgado88].

Le filtre des moindres carrés récurrents EFRA est donné sous la forme suivante:

$$\theta_k = \theta_{k-1} + \frac{\alpha P_k \Phi_k}{1 + \Phi_k^T P_k \Phi_k} e_k \quad (3-23)$$

$$e_k = y_k - \Phi_k^T \theta_{k-1} \quad (3-24)$$

$$P_k = \frac{1}{\lambda} P_{k-1} - \frac{\alpha P_{k-1} \Phi_k \Phi_k^T P_{k-1}}{1 + \Phi_k^T P_{k-1} \Phi_k} + \beta I - \delta P_{k-1}^2 \quad (3-25)$$

On peut faire une analogie avec le filtre de Kalman. On remarque que la matrice d'autocorrélation $R(k)$ dans la forme du RLS est unitaire, que le vecteur de transition $C(k)$ correspond à Φ^T . Ce qui donne le filtre de Kalman avec facteur d'oubli et mise à la valeur initiale de la forme suivante:

$$\hat{x}(k) = \hat{x}(k-1) + \alpha K(k)[y(k) - C(k)\hat{x}(k-1)] \quad (3-26)$$

$$K(k) = \tilde{P}(k-1)C^T(k)[C(k)\tilde{P}(k-1)C^T(k) + R(k)]^{-1} \quad (3-27)$$

$$\tilde{P}(k) = \left[\frac{I}{\lambda} - \alpha K(k)C(k) \right] \tilde{P}(k-1) + \beta I - \delta \tilde{P}(k-1)^2 \quad (3-28)$$

Dans l'article, une démonstration mathématique donne des conditions sur les facteurs α , β , γ et δ de façon que l'estimation des paramètres soit la plus fidèle.

$$0 < \gamma < \alpha < 1 \quad (3-29)$$

$$(\gamma - \alpha)^2 + 4\beta\delta < (1 - \alpha)^2 \quad (3-30)$$

$$\beta > 0 \quad \delta > 0 \quad (3-31)$$

$$\sigma I \leq P_0 \leq \nu I \quad (3-32)$$

Avec $\gamma = \frac{1-\lambda}{\lambda}$, P_0 la matrice initiale de covariance, les facteurs ν et σ sont donnés ci-dessous:

$$\nu = \frac{\gamma}{2\delta} \left[1 + \left(1 + \frac{4\beta\delta}{\gamma^2} \right)^{1/2} \right] \quad (3-33)$$

$$\sigma = \left(\frac{\alpha - \gamma}{2\delta} \right) \left[-1 + \left(1 + \frac{4\beta\delta}{(\alpha - \gamma)^2} \right)^{1/2} \right] \quad (3-34)$$

Dans le cadre de la commande, les facteurs sont souvent situés dans les intervalles suivants:

α ajuste le gain de Kalman, typiquement $\alpha \in [0,1 ; 0,5]$

β est la petite constante qui est directement reliée à la valeur propre minimum de P , typiquement $\beta \in [0 ; 0,01]$

δ est la petite constante qui est inversement reliée à la valeur propre maximum de P , typiquement $\delta \in [0 ; 0,01]$

λ est le facteur d'oubli exponentiel, typiquement $\lambda \in [0,9 ; 0,99]$

P_0 est une matrice diagonale de terme suffisamment élevé.

Remarque : pour la simulation, il est important de vérifier les conditions sur le choix des facteurs et autoriser l'estimation des coefficients de la tangente seulement si les conditions 3-29 à 3-32 sont vérifiées.

3.2.3 Simulation

Avant de passer à la phase d'expérimentation avec implantation de l'algorithme dans une structure matérielle, il est important de générer des données synthétiques. La disposition des deux transducteurs de l'article [Song96] est utilisée, mais ces deux derniers vont être

fixés sur le robot au lieu d'être rotatifs. Le système mobile va effectuer des aller-retours en direction des murs de différentes inclinaisons qui sont comparables à différentes équations de la tangente. Pour cela, il est important d'avoir deux référentiels, un référentiel mobile $\{M\}$ qui se trouve à mi-parcours entre les deux transducteurs pour la mesure de distance et un autre fixe appelé référentiel origine $\{O\}$ qui donnera l'équation dans un repère global. Il est nécessaire d'effectuer un changement de référentiel. La théorie sur le changement de référentiel provenant du livre de [Craig89] est présentée en annexe C, sous la forme d'un petit rappel.

Par un changement de repère (rotation de référentiel), il est possible de simuler différentes inclinaisons du mur à partir d'une génération de points de mesure pour un seul type de mur, un mur vertical. Pour rendre encore plus réaliste le simulateur, il est possible que le système sensoriel détecte plusieurs types de mur dans une même simulation, en changeant l'inclinaison du mur. Au début de chaque simulation, l'utilisateur définit le nombre de murs que le système mobile va détecter ainsi que le nombre de fois que le système va détecter un même mur. L'ensemble des modules du simulateur est expliqué de façon détaillée dans l'annexe C.

3.2.3.1 Calcul des termes du filtre de Kalman pour un changement de repère

Les seuls termes qui changent lorsque l'on effectue le changement de repère, sont ceux qui proviennent directement de l'équation de droite $\mathbf{p}^*x_M + \mathbf{q}^*y_M - 1 = 0$, car x et y sont affectées par le changement de repère. Ces termes sont donc ceux qui comportent des dérivées partielles : $\mathbf{C}(\mathbf{k})$, $\mathbf{y}(\mathbf{k})$ et $\mathbf{R}(\mathbf{k})$.

3.2.3.2 Fonction non linéaire après le changement de repère

La fonction non linéaire $\mathbf{f}(\mathbf{a}, \mathbf{b}) = 0$ s'écrit sous la forme suivante

$$\mathbf{f}(\mathbf{a}, \mathbf{b}) = \mathbf{p} \left(\frac{\mathbf{d}_1 + \mathbf{d}_2}{2 \cos \psi} \cos(\theta + \gamma) + \mathbf{x}_p \right) + \mathbf{q} \left(\frac{\mathbf{d}_1 + \mathbf{d}_2}{2 \cos \psi} \sin(\theta + \gamma) + \mathbf{y}_p \right) - 1 = 0 \quad (3-35)$$

Cette équation est obtenue simplement en utilisant les règles inverses d'addition des fonctions trigonométriques, $\sin(g+h)$ et $\cos(g+h)$. Le changement de repère est expliqué dans [Craig89], les variables \mathbf{x}_p et \mathbf{y}_p représentent les coordonnées du référentiel mobile dans celui global.

3.2.3.3 Calcul des dérivées partielles

$$\left. \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right|_{\mathbf{a}_0, \mathbf{b}_0} = \frac{(\mathbf{d}_{10} + \mathbf{d}_{20}) \cos(\theta_0 + \gamma_0)}{2 \cos \left\{ \sin^{-1} \left(\frac{|\mathbf{d}_{10} - \mathbf{d}_{20}|}{s_d} \right) \right\}} + \mathbf{x}_p = \frac{(\mathbf{d}_{10} + \mathbf{d}_{20}) \cos(\theta_0 + \gamma_0)}{2 * \mathbf{D}} + \mathbf{x}_p \quad (3-36)$$

$$\left. \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \right|_{\mathbf{a}_0, \mathbf{b}_0} = \frac{(\mathbf{d}_{10} + \mathbf{d}_{20}) \sin(\theta_0 + \gamma_0)}{2 \cos \left\{ \sin^{-1} \left(\frac{|\mathbf{d}_{10} - \mathbf{d}_{20}|}{s_d} \right) \right\}} + \mathbf{y}_p = \frac{(\mathbf{d}_{10} + \mathbf{d}_{20}) \sin(\theta_0 + \gamma_0)}{2 * \mathbf{D}} + \mathbf{y}_p \quad (3-37)$$

$$\left. \frac{\partial \mathbf{f}}{\partial \mathbf{d}_{1,2}} \right|_{\mathbf{a}_0, \mathbf{b}_0} = \frac{\partial \left(\mathbf{p} \cdot \frac{\mathbf{d}_1 + \mathbf{d}_2}{2 * \mathbf{D}} \cos(\theta + \gamma) + \mathbf{q} \cdot \frac{\mathbf{d}_1 + \mathbf{d}_2}{2 * \mathbf{D}} \sin(\theta + \gamma) \right)}{\partial \mathbf{d}_{1,2}} \bigg|_{\mathbf{a}_0, \mathbf{b}_0} \quad (3-38)$$

$$\left. \frac{\partial \mathbf{f}}{\partial \theta_0} \right|_{\mathbf{a}_0, \mathbf{b}_0} = -\mathbf{p}_0 \frac{(\mathbf{d}_{10} + \mathbf{d}_{20}) \sin(\theta_0 + \gamma_0)}{2 * \mathbf{D}} + \mathbf{q}_0 \frac{(\mathbf{d}_{10} + \mathbf{d}_{20}) \cos(\theta_0 + \gamma_0)}{2 * \mathbf{D}} \quad (3-39)$$

Pour les dérivées partielles de \mathbf{d}_1 et \mathbf{d}_2 , on remarque au dénominateur $2 * \mathbf{D}$ des fonctions emboîtées $g[h(x)] = goh$, et sa dérivée correspond à $g'(h) * h'$.

On rappelle que la dérivée de $\sin^{-1}(\mathbf{u}) = \frac{\mathbf{u}'}{\sqrt{1 - \mathbf{u}^2}}$.

On peut remarquer que les valeurs absolues risquent de poser divers cas pour le calcul des dérivées partielles. Les valeurs absolues peuvent être décomposées par le produit d'une fonction **signe(u)** multipliée par le contenu de la valeur absolue **u**. Où **signe(u)** = 1 si l'expression de la valeur absolue est positive, **signe(u)** = -1 si l'expression de la valeur absolue est négative.

$$\left. \frac{\partial(2 * D)}{\partial d_1} \right|_{a_0, b_0} = \frac{\partial \left(2 \cos \left\{ \sin^{-1} \left(\frac{\text{signe}(d_1 - d_2) * (d_1 - d_2)}{s_d} \right) \right\} \right)}{\partial d_1} \bigg|_{a_0, b_0} \quad (3-40)$$

$$\left. \frac{\partial(2 * D)}{\partial d_1} \right|_{a_0, b_0} = -2 * \left[\sin \left(\sin^{-1} \left(\frac{\text{signe}(d_{10} - d_{20}) * (d_{10} - d_{20})}{s_d} \right) \right) * \frac{\frac{\text{signe}(d_{10} - d_{20})}{s_d}}{\sqrt{1 - \frac{(d_{10} - d_{20})^2}{s_d^2}}} \right] \quad (3-41)$$

$$\left. \frac{\partial(2 * D)}{\partial d_2} \right|_{a_0, b_0} = - \left(\left. \frac{\partial(2 * D)}{\partial d_1} \right|_{a_0, b_0} \right) = \frac{2 * (d_{10} - d_{20})}{s_d \sqrt{s_d^2 - (d_{10} - d_{20})^2}} \quad (3-42)$$

En posant une variable locale pcqsD

$$\text{pcqsD} = \frac{(p * \cos(\theta_0 + \gamma_0) + q * \sin(\theta_0 + \gamma_0))}{2 * D} \quad (3-43)$$

On obtient :

$$\frac{\partial f(a, b)}{\partial d_1} = \text{pcqsD} - \frac{\frac{\partial f(2 * D)}{\partial d_1} * (d_{10} + d_{20}) * \text{pcqsD}}{(2 * D)} \quad (3-44)$$

$$\frac{\partial f(a, b)}{\partial d_1} = \text{pcqsD} \left(1 + \frac{(d_{10}^2 - d_{20}^2) * \text{pcqsD}}{(s_d * D * \sqrt{s_d^2 - (d_{10} - d_{20})^2})} \right) \quad (3-45)$$

On peut en déduire la dérivée partielle par rapport à d_2

$$\frac{\partial f(a, b)}{\partial d_2} = pcqsD \left(1 - \frac{(d_{10}^2 - d_{20}^2) * pcqsD}{s_d * D * \sqrt{s_d^2 - (d_{10} - d_{20})^2}} \right) \quad (3-46)$$

3.2.3.4 Réduction du nombre d'opérations

Dans la série de réductions du nombre de calculs, on peut remarquer que la dérivée partielle (3-45) par rapport à θ ou $b(3)$, peut être écrite sous la forme suivante:

$$\left. \frac{\partial f}{\partial \theta_0} \right|_{(a_0, b_0)} = \text{fonction}(p, q, \frac{\partial f}{\partial d_1}, \frac{\partial f}{\partial d_2}) = -p * \left. \frac{\partial f}{\partial d_1} \right|_{(a_0, b_0)} + q * \left. \frac{\partial f}{\partial d_2} \right|_{(a_0, b_0)} \quad (3-47)$$

De même, les dérivées partielles par rapport à $b(1)$ et $b(2)$ sont identiques à un signe près, il suffit de calculer les deux termes et pour la dérivée partielle à $b(1)$, on soustrait les deux termes et pour la dérivée partielle à $b(2)$, on additionne les deux termes. On rappelle que les vecteurs $b(1)$ et $b(2)$ correspondent respectivement aux vecteurs $d(1)$ et $d(2)$.

3.2.3.5 Simulation

L'ensemble de l'algorithme décrit ci-dessus va être implanté sous Matlab et est testé avec différents types d'environnements qui sont plus ou moins bruités. En tenant compte du problème d'observabilité, il ne faut pas prendre le cas de l'estimation d'un mur vertical à une distance d du robot, car il s'agit de la seule combinaison où le filtre de Kalman sans facteurs d'oubli donne les bons estimés. Il est important de simuler un cas où il y a plusieurs inclinaisons de mur en testant le cas où l'on passe d'un mur vertical (coefficient $q = 0$) à un mur incliné (coefficient $q \neq 0$).

3.2.3.6 Simulation et choix des paramètres de l'algorithme de fusion

Dans les sections suivantes, la version de l'algorithme de Kalman avec facteurs d'oubli est donnée. Ce filtre est appliqué à l'estimation de paramètres d'un segment de mur, provenant de données ultrasonores. Il est important pour le filtre de Kalman d'avoir un bon estimé du bruit. La variance du bruit de notre système est estimée comme étant la variance associée à la précision des données reçues. Le calcul de la distance par la mesure du temps de vol est déterminé par $d(\text{mesurée}) = d(\text{calculée}) \pm \Delta d$, avec Δd la résolution du système. La résolution d'un système de mesure de distance courant est d'environ 1 cm, ce qui donne $\Delta d = 0.5$ cm. La distribution de l'erreur qui est distribuée uniformément peut s'approcher par une loi gaussienne d'écart type Δd , ce qui correspond à une variance de $25 \cdot 10^{-6} \text{ m}^2$. Pour la simulation, on choisit une variance de $2 \cdot 10^{-5} \text{ m}^2$. Il ne reste plus qu'à ajuster les cinq facteurs d'oubli et de simuler notre système avec différents niveaux de bruit. On peut s'attendre au résultat suivant avec le filtre de Kalman en fonction du bruit imposé : si on se trouve en sur estimation de la variance (les données de mesure sont bruitées avec une variance plus faible que celle des données dans le filtre de Kalman) l'estimation des paramètres ne va pas être affectée par cette sur évaluation; par contre, si on se trouve en sous évaluation du bruit, l'estimation des paramètres va être affectée et l'estimation risque de suivre le bruit.

La méthode pour l'ajustement des paramètres est la suivante, on ajuste grossièrement les paramètres du modèle sans bruit, on peut utiliser plusieurs itérations du filtre sur les mêmes mesures. Cependant, le risque de cette méthode est d'estimer les paramètres pour un mur avec bruit si on effectue trop d'itérations. Pour valider ce choix des paramètres, on va ajouter aux données de mesure, du bruit de différentes variances. Le bruit est ajouté aux données de mesure tel qu'illustré à la figure 3-11. Un enregistrement du bruit, de même longueur que le vecteur de mesure \mathbf{b} , est appliqué à tous les algorithmes présentés.

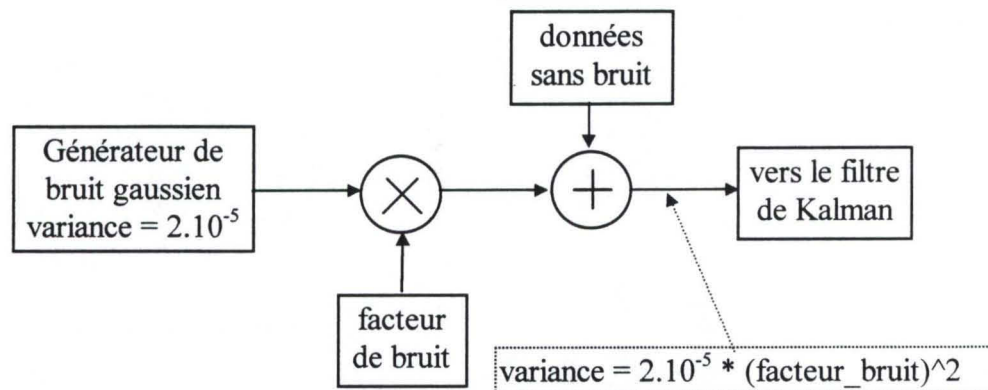


Figure 3-11 : Générateur de bruit

Remarque : on utilise un bruit blanc gaussien de variance 2.10^{-5} m^2 . Ces propriétés statistiques donnent que 2/3 des points sont compris entre \pm le niveau de variance, 95% des points dans la plage ± 2 fois le niveau de la variance et 99.7% pour ± 3 fois ce niveau.

Pour l'ensemble des figures représentant les résultats de simulations, elles sont toutes composées de la réponse théorique sans bruit (trait plein) et de la réponse par la résolution d'un système bruité de 2 équations à 2 inconnues (traits alternés). Les réponses des simulations présentées sont ajoutées à cette figure type et sont identifiées avec un trait continu marqué de symboles spécifiques.

Tous les algorithmes exposés ont leur programme respectif (code Matlab) à l'annexe C.

La figure 3-12 donne l'estimation des 2 coefficients de l'équation de la tangente pour la version de l'algorithme de Kalman proposée par l'article [Song96] (symbole o; programme: kalman_song.m), pour la version de cet algorithme avec facteurs d'oubli adapté du RLS_EFRA de [Salgado88] (symbole *; programme: kalman_song_salgado.m).

Tableau 3-3: Paramètres de simulation de la figure 3-12

Algorithme	Song96	Song96 + EFRA
nb_ité	1	15
P_0	$20 \cdot I_{2 \times 2}$	$20 \cdot I_{2 \times 2}$
α	\emptyset	0,3
β	\emptyset	0,5
δ	\emptyset	0,0005
λ	\emptyset	0,85

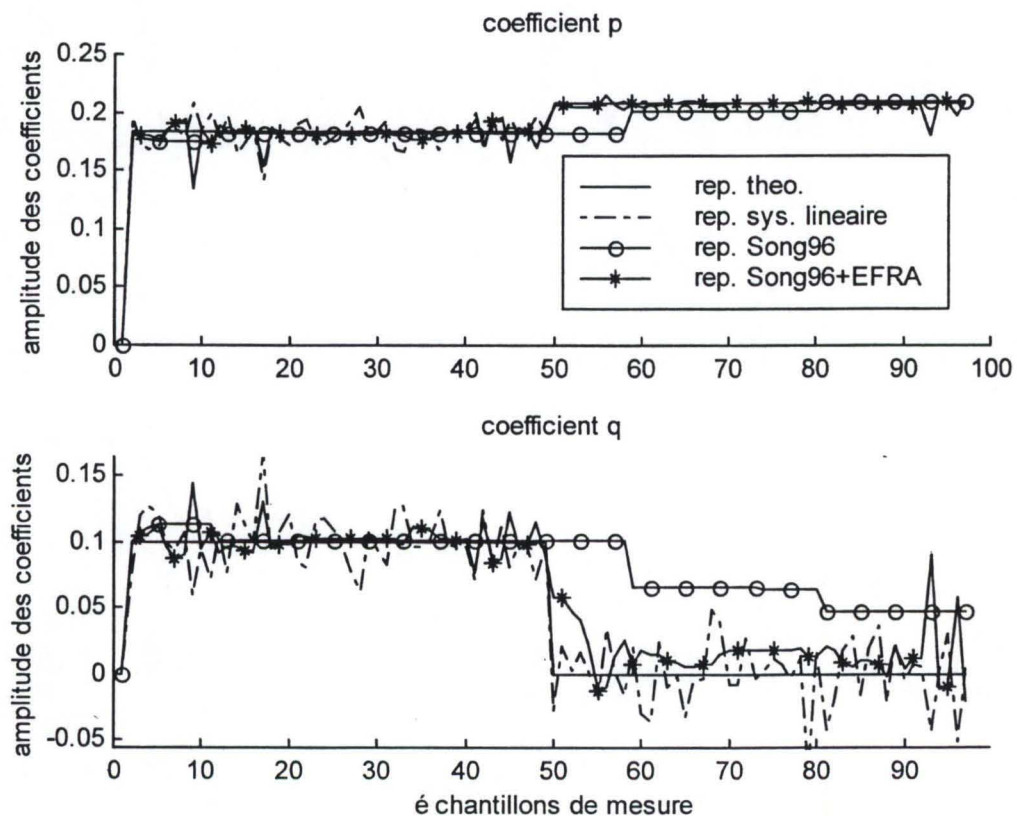


Figure 3-12 : Estimations à l'aide du filtre de Kalman étendu

La réponse du filtre de Kalman proposé par [Song96] a des problèmes de convergence, ces derniers sont prévisibles à cause de la non observabilité du système. L'amélioration apportée pour rendre le système observable porte ses fruits puisque la réponse converge

correctement. Le filtre de Kalman présente une certaine dynamique qui doit éviter de varier trop aux impulsions de bruit, cette dynamique est présente mais l'algorithme a tendance à amplifier le bruit. La principale raison de ce bruit provient de la linéarisation qui est effectuée à l'aide de points d'opération bruités. Pour réduire le bruit, il est nécessaire de contourner la linéarisation.

3.2.3.7 Recherche d'un nouvel algorithme

En observant l'équation (3-20), on remarque que cette dernière est déjà linéaire ($\mathbf{p} \cdot \mathbf{x} + \mathbf{q} \cdot \mathbf{y} - 1 = 0$), les termes à estimer sont les coefficients (\mathbf{p}, \mathbf{q}). Les équations (3-48) et (3-49) montrent une nouvelle façon d'écrire le système.

$$\mathbf{p} \cdot \mathbf{x} + \mathbf{q} \cdot \mathbf{y} - 1 = 0 \quad (3-48)$$

$$[\mathbf{x} \ \mathbf{y}] * [\mathbf{p} \ \mathbf{q}]^T = 1 \quad (3-49)$$

Cette notation ressemble à l'équation de base de l'algorithme RLS ou algorithme récursif à moindres carrés.

$$\mathbf{C}^T = [\mathbf{x} \ \mathbf{y}], \ \mathbf{x} = [\mathbf{p}, \mathbf{q}]^T, \ y_{\text{mes}} = 1 \quad (3-50)$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{K}_k * (y_{\text{mes}} - \mathbf{C}^T * \mathbf{x}_{k-1}) \quad (3-51)$$

$$\mathbf{K}_k = \frac{\mathbf{P}_{k-1} * \mathbf{C}_k}{\mathbf{I} + \mathbf{C}_k^T * \mathbf{P}_{k-1} * \mathbf{C}_k} \quad (3-52)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k * \mathbf{C}_k^T) * \mathbf{P}_{k-1} \quad (3-53)$$

On définit l'erreur comme l'erreur d'estimation puisque la valeur mesurée (y_{mes}) qui est normalement bruitée, est constante à 1. Pour éviter les problèmes de mauvaise convergence, on utilise l'algorithme RLS avec un facteur d'oubli λ (3-54 et 3-55). Le rôle de ce terme est d'assurer que les données du passé lointain sont oubliées afin de suivre les

variations statistiques des observations quand le filtre opère dans un environnement non stationnaire.

$$\mathbf{K}_k = \frac{\mathbf{P}_{k-1} * \mathbf{C}_k}{\lambda \mathbf{I} + \mathbf{C}_k^T * \mathbf{P}_{k-1} * \mathbf{C}_k} \quad (3-54)$$

$$\mathbf{P}_k = (\mathbf{I}/\lambda - \mathbf{K}_k * \mathbf{C}_k^T) * \mathbf{P}_{k-1} \quad (3-55)$$

La figure 3-13 présente les résultats obtenus avec les 2 algorithmes de type RLS, celui sans facteur d'oubli (symbole o; programme: rls_simple.m) où la convergence n'est pas garantie et celui avec un facteur d'oubli (symbole *; programme: rls_lambda.m) qui doit converger correctement.

Tableau 3-4: Paramètres de simulation de la figure 3-13

Algorithme	RLS	RLS + oubli
nb_ité	1	4
P₀	20*I _{2*2}	20*I _{2*2}
λ	∅	0.95

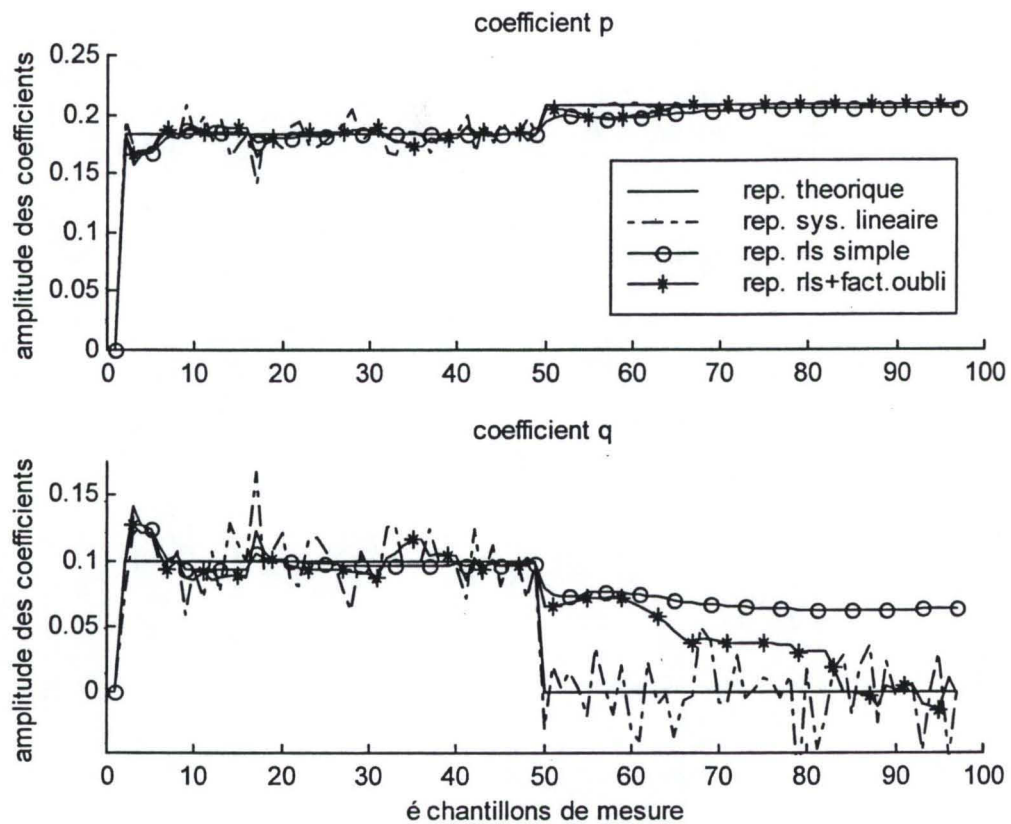


Figure 3-13 : RLS avec ou sans facteur d'oubli

Les réponses obtenues à la figure 3-13 montrent les limites de cet identificateur simple en terme de vitesse de convergence malgré la présence d'un facteur d'oubli valant 0,95. Des essais avec des valeurs plus faibles permettent d'oublier plus rapidement le passé, mais ils présentent l'inconvénient d'être plus sensibles aux bruits. Plusieurs méthodes existent pour accélérer la convergence avec des algorithmes spécifiques. Deux d'entre-elles sont retenues:

- L'algorithme de [Salgado88] avec sa structure EFRA qui permet de limiter les valeurs propres de la matrice de covariance \mathbf{P} ;
- Un algorithme à initialisation périodique de la matrice \mathbf{P} pour un système qui varie brutalement dans le temps (le cas ici) comme proposé dans le livre de [Astrom95].

La version de l'algorithme RLS avec facteurs d'oubli et *resetting* exponentiel [Salgado88] est celle qui a servi de base à la création de l'algorithme Song96 + EFRA (paragraphe

3.2.2.4). Les principales équations sont données en 3-56 à 3-58.

$$\theta_k = \theta_{k-1} + \frac{\alpha \mathbf{P} \Phi_k}{1 + \Phi_k^T \mathbf{P} \Phi_k} \mathbf{e}_k \quad (3-56)$$

$$\mathbf{e}_k = \mathbf{y}_k - \Phi_k^T \theta_{k-1} \quad (3-57)$$

$$\mathbf{P}_k = \frac{1}{\lambda} \mathbf{P}_{k-1} - \frac{\alpha \mathbf{P}_{k-1} \Phi_k \Phi_k^T \mathbf{P}_{k-1}}{1 + \Phi_k^T \mathbf{P}_{k-1} \Phi_k} + \beta \mathbf{I} - \delta \mathbf{P}_{k-1}^2 \quad (3-58)$$

La version à réinitialisation périodique présente l'avantage d'être simple (ajout d'un compteur à la version RLS) et efficace (convergence rapide des premiers échantillons, voir la figure 3-13). La variable `t_reset` contient le nombre de mesures entre chaque initialisation. La figure 3-14 présente ces deux algorithmes avec initialisation périodique ou *reset* (symbole 0; programme: `rls_salgado.m`) et avec la structure EFRA (symbole *; programme: `rls_reset.m`).

Tableau 3-5 : Paramètres de simulation de la figure 3-14

Algorithme	RLS + reset	RLS + EFRA
nb_ité	5	10
P₀	20*I _{2*2}	20*I _{2*2}
λ	∅	0,80
α	∅	0,60
β	∅	0,1
δ	∅	0,005
t_reset	5	∅

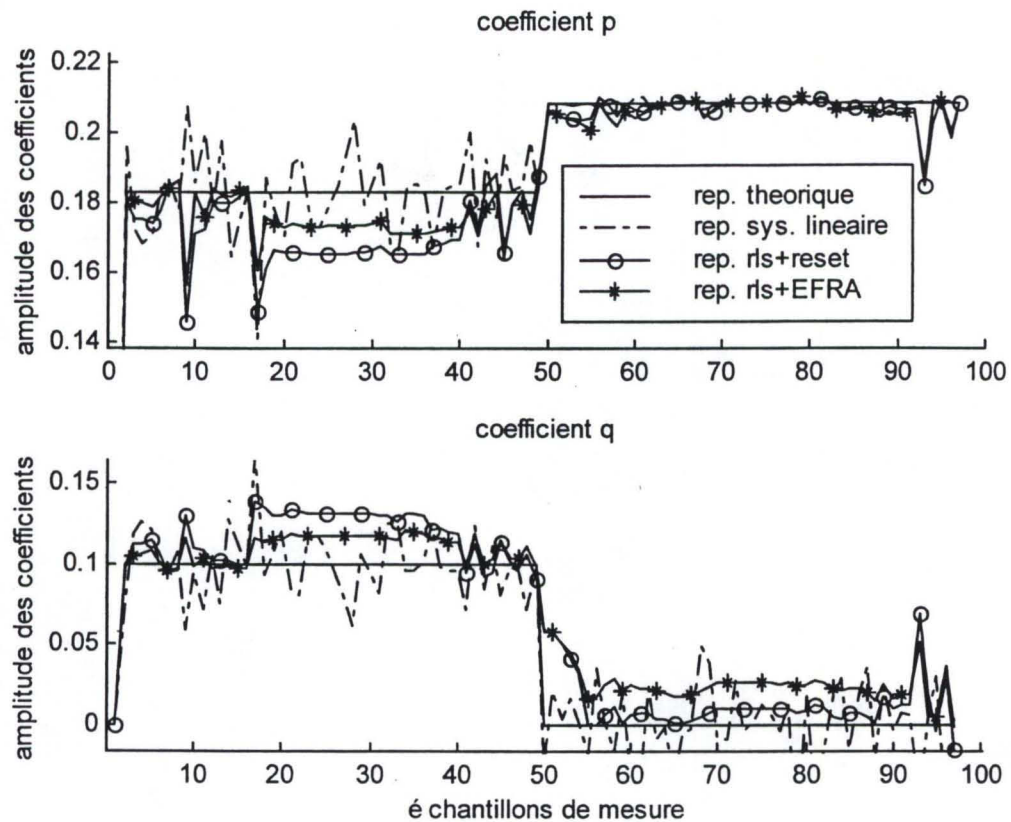


Figure 3-14 : RLS avec facteurs d'oubli et resetting exponentiel

Les résultats obtenus par les deux algorithmes sont globalement bons et proches.

- L'algorithme RLS à réinitialisation périodique converge très rapidement à chaque variation du milieu, il est un peu plus sensible au bruit que l'algorithme EFRA. On remarque que l'erreur de position est importante à la mesure 18, il s'agit d'une réinitialisation périodique alors que les paramètres théoriques ne varient pas, le ratio entre la variation du bruit et l'erreur de position est beaucoup plus grand que celui lors du changement brusque d'environnement. Dans les différentes simulations non montrées, on remarque que si l'initialisation est effectuée trop souvent, on ajoute un délai à la convergence; si on augmente la valeur initiale de P_0 , on augmente la rapidité de la convergence et aussi le niveau de bruit.

- Pour algorithme RLS avec la structure EFRA, les paramètres estimés fluctuent peu aux bruits de mesure, convergent rapidement aux changements d'environnement mais une erreur de position diminuant légèrement est présente. Les résultats par rapport au RLS avec un facteur d'oubli (figure 3-13) sont excellents en terme de convergence et de sensibilité au bruit.

Chaque algorithme présente ses avantages, l'algorithme RLS à réinitialisation périodique converge très rapidement en cas de changement brusque avec réinitialisation, l'algorithme RLS à structure EFRA est adaptatif en limitant les valeurs propres de la matrice de covariance \mathbf{P} pour assurer un suivi des paramètres théoriques.

Je propose de développer mon propre algorithme en fusionnant les deux structures. Le *reset* déclenché par une variation du paramètre estimé remplace le *reset* périodique. Je choisis une variation de 10% du coefficient (variable seuil). On remarque que la réponse est invariante lors du *reset* ce qui peut ralentir la convergence si elle est déclenchée fréquemment. Pour remédier à ce problème, j'ajoute une petite constante, un biais, correspondant à un niveau supérieur du bruit (variable biais). Dans les zones de variations brusques des coefficients, la convergence du système est retardée par des réinitialisations répétitives de \mathbf{P}_0 . J'ajoute une immunité (un nombre de mesures) où le système ne peut pas être réinitialisé (variable immu). Ce filtre est appelé RLS_IMCEVI qui est une abréviation d'un algorithme RLS avec Initialisation de la Matrice de Covariance pour des Estimés Variants brusquement et possédant une Immunité (symbole o; programme: rls_imcevi.m).

Tableau 3-6 : Paramètres de simulation de la figure 3-15

Algorithme	RLS_IMCEVI
nb_ité	1
P_0	$5 \cdot I_{2 \times 2}$
λ	0,96
seuil	10%
immu	15
biais	0,015

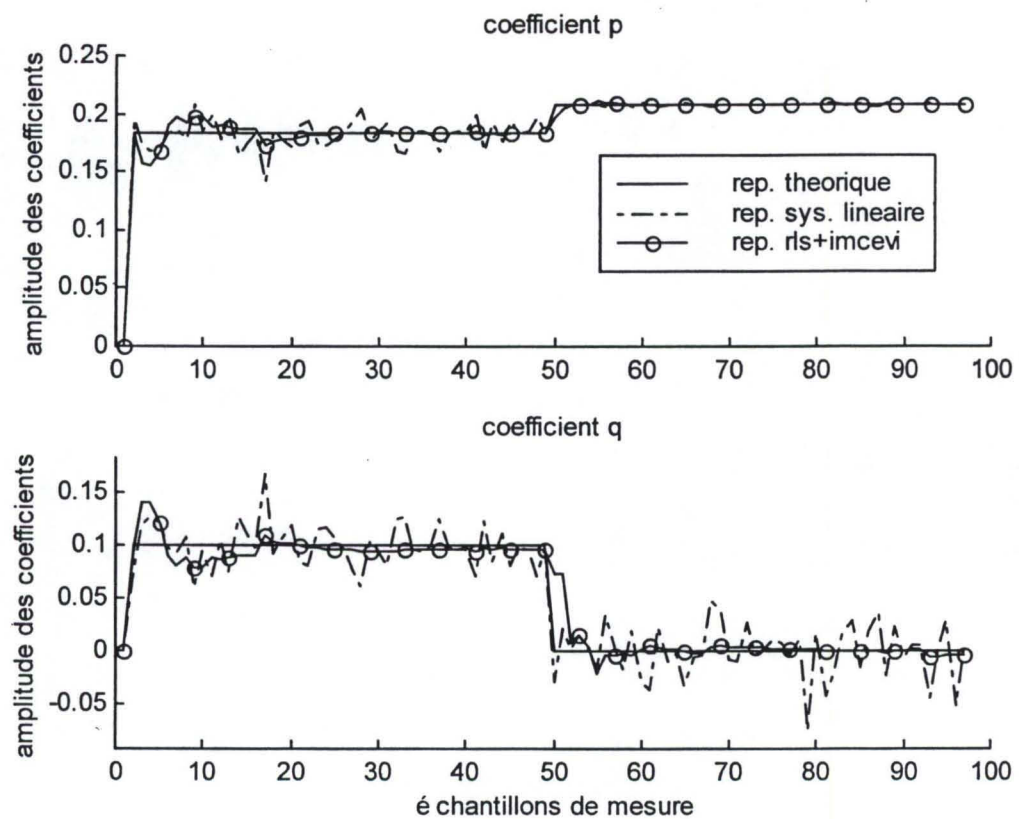


Figure 3-15 : RLS_IMCEVI appliqué à l'estimation des coefficients de la tangente

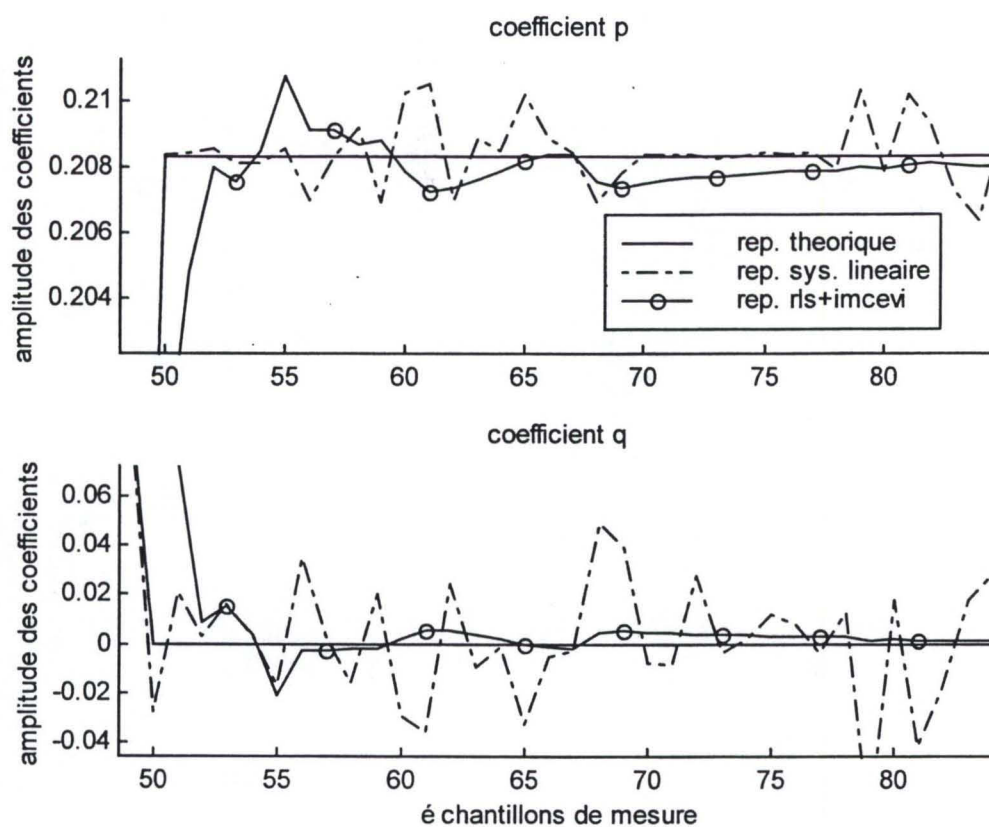


Figure 3-16 : Détails de l'estimation avec le RLS_IMCEVI

Les réponses fournies par le filtre appelé RLS_IMCEVI (figure 3-15 et 3-16) convergent rapidement en moins de 10 points de mesures et leur niveau de bruit est inférieur à celui obtenu par le modèle géométrique bruité. Cette variante du RLS est le meilleur choix testé et demande de faibles ressources logicielles : pas d'itération et une structure moins complexe que celle du filtre de Kalman.

Dans ce chapitre, deux algorithmes de reconstitution sont présentés. L'étude effectuée des paramètres de l'algorithme de mesure de distance permet d'obtenir une bonne résolution. L'estimation des coefficients de l'équation de la tangente est réalisée par une nouvelle approche : le système qui décrit le modèle de la géométrie est linéaire; l'algorithme RLS est adapté aux comportements brutaux des données ultrasonores à l'aide des moyens suivants : réinitialisation de la matrice \mathbf{P} en cas de variations importantes des coefficients, plage d'immunité à la réinitialisation.

Chapitre 4 :

Schéma et implantation

Dans les chapitres précédents, la mesure de distance par le temps de vol est expliquée depuis l'élément sensoriel qui émet et reçoit les ondes, jusqu'au traitement de l'écho. Dans le chapitre 2, une présentation du kit Polaroid donne une idée sur l'interface pour obtenir les échantillons de mesure. Ici, cette interface sera conçue de façon à incorporer l'algorithme de mesure dans un micro-contrôleur qui sera préalablement choisi.

4.1 Étude du bloc de conversion

4.1.1 Principe

Le bloc de conversion tel qu'il a été défini dans le chapitre 2.1 et 2.5, est chargé de la conversion des signaux sensoriels vers et provenant de la partie numérique, la reconstitution. Dans ce grand bloc, il peut être décomposé en deux unités fonctionnelles, l'excitation et la conversion. Ce bloc communique avec l'élément sensoriel, le transducteur et la partie matérielle qui héberge l'algorithme de mesure de distance dont l'organigramme se trouve à la figure 3-4.

4.1.2 Analyse fonctionnelle

4.1.2.1 Excitation

L'excitation envoie un train d'onde pour exciter le transducteur, il faut que le signal qui excite le transducteur soit suffisamment important pour que l'élément sensoriel puisse

générer une onde acoustique sans altérer les performances du transducteur. Ici, le transducteur utilisé est un élément électrostatique de chez Polaroid, *600 Series*. Les données techniques pour l'excitation se trouvent en annexe H.

L'unité fonctionnelle excitation est représentée par un schéma-bloc, figure 4-1.

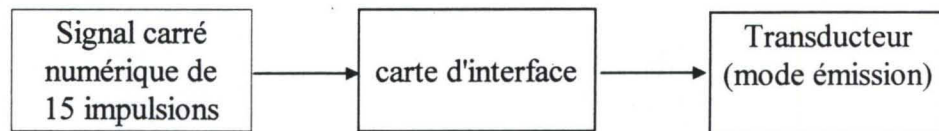


Figure 4-1 : Schéma fonctionnel de l'excitation

4.1.2.2 Conversion

Le module de conversion reçoit un signal électrique image de la pression acoustique reçue par le transducteur et doit le mettre en forme de façon à obtenir une amplitude maximale constante à l'entrée du convertisseur A/N. Cette amplitude doit être comprise dans la plage d'entrée du convertisseur.

L'unité fonctionnelle conversion est représentée par un schéma-bloc, figure 4-2.

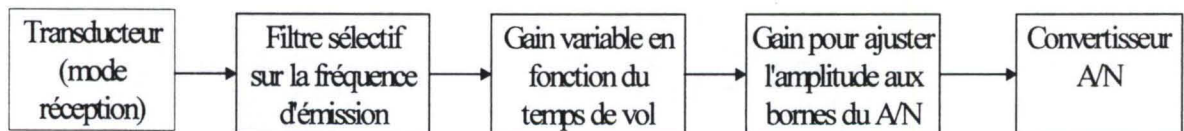


Figure 4-2 : Schéma fonctionnel de la conversion

4.1.3 Analyse structurelle

Dans ce paragraphe, on va proposer les montages retenus pour la carte qui interface le

transducteur au bloc de reconstitution. Les documents techniques du kit Polaroid sont donnés dans l'annexe H.

4.1.3.1 Excitation

L'interface sur la carte Polaroid

Le TL851 fournit un signal XMIT qui est un train d'ondes carrées à la fréquence de résonnance du transducteur, figure 4-3. Dans les spécifications techniques du TL851, cette sortie peut fournir jusqu'à 140 mA. Cette valeur est importante pour un circuit qualifié de digital, nous avons effectué cette mesure en insérant une résistance de court-circuit (de très faible valeur) entre la sortie XMIT et la base du transistor afin de ne pas modifier le potentiel de base. Le courant maximal mesuré est 160 mA.

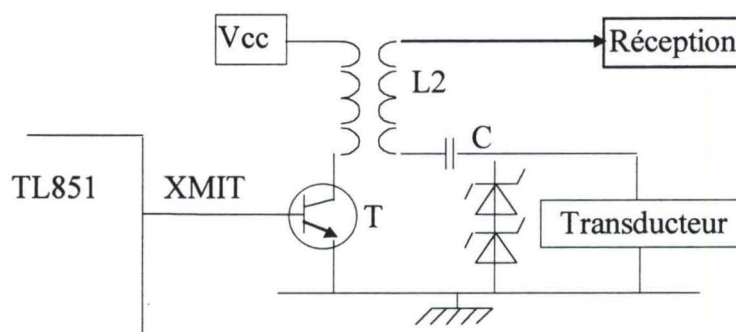


Figure 4-3 : Schéma d'excitation (d'après Polaroid)

Ce courant circulant dans la base fournit un courant de collecteur supérieur à l'ampère. Ce courant est impulsionnel et passe dans une bobine, qui correspond au primaire du transformateur. La tension aux bornes de la bobine est $U_L = \frac{L \cdot di(t)}{dt}$, comme la pente de la commutation est très élevée, la tension aux bornes du secondaire pourra atteindre une tension crête à crête de 300 volts à cause du couplage entre le primaire et le secondaire (coefficient d'induction mutuelle). Ce niveau de tension est nécessaire pour exciter la

structure mécanique du transducteur afin d'émettre une onde acoustique. Cependant le transducteur devra être polarisé par un signal positif. Un condensateur se chargera du découplage et deux diodes zeners (2×160 Volts) protégeront le transducteur contre les surtensions.

L'interface proposée

Comme l'utilisation d'un micro-contrôleur pourrait suffire pour implanter l'algorithme de mesure de distance, il servira de séquenceur pour l'ensemble des circuits d'interface. Le micro-contrôleur devrait délivrer le train d'ondes carrées pour exciter le transducteur comme le faisait le TL851. Cependant, le micro-contrôleur qui a des sorties compatibles TTL, ne fournit pas la même quantité de courant que le TL851, il est nécessaire d'insérer un étage d'amplification.

Le circuit d'interface entre le micro-contrôleur et le signal d'excitation (XMIT ou la base du transistor) a comme fonction de générer 160 mA de courant sous 1.1 Volt. Toutefois la sortie du micro-contrôleur qui est habituellement un collecteur ouvert, peut absorber jusqu'à 5 mA. Il faudra choisir une structure Darlington pour effectuer le gain en courant. Cette dernière est calculée et justifiée dans l'annexe D-1

4.1.3.2 Conversion

L'interface du module Polaroid

La méthode pour évaluer le temps de vol chez Polaroid est une détection d'enveloppe à l'aide d'un seuil de basculement. Cette détection englobe le gain variable pour compenser l'atténuation de l'écho en fonction du temps de vol. Cette fonction est effectuée dans le circuit le TL852, le gain variable est effectué à l'aide d'une amplification par palier, se reporter au tableau 2-3, et il est temporellement commandé par le TL851 qui sert à l'excitation. Après la détection d'enveloppe effectuée, un signal numérique arrête le

compteur qui a été initialisé au début de l'émission, pour connaître la durée du temps de vol.

L'interface proposée

Le chapitre 3 a montré la performance de la méthode d'intercorrélation. Cette dernière utilise la valeur absolue de l'écho. Il est relativement simple de calculer la valeur absolue logiciellement à condition de convertir l'onde reçue dans la plage du convertisseur (partie positive et négative). Il est important de ne pas altérer la forme du signal reçu tout en éliminant le bruit de mesure à l'aide d'un filtre sélectif.

Les signaux sont filtrés par un filtre passe-bande de Butterworth composé par deux filtres du deuxième ordre montés en cascade (un passe bas suivi d'un passe haut) puis amplifiés pour compenser l'atténuation provenant des filtres. La structure de Butterworth offre l'avantage vis-à-vis des structures de Rauch et Sallen-Kay d'avoir une contre réaction unitaire, ce qui évite les problèmes liés à la bande passante à cause d'une contre réaction élevée.

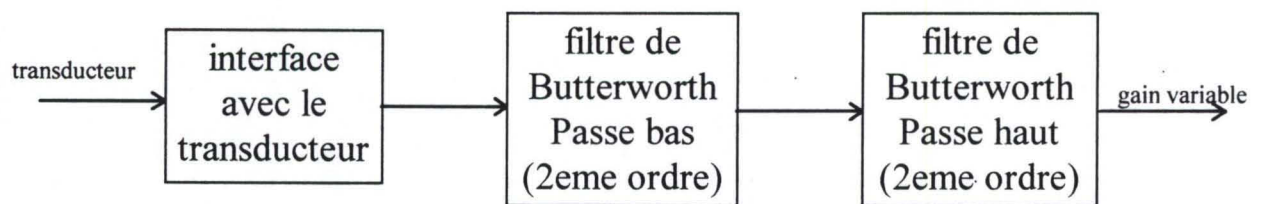


Figure 4-4 : Filtre sélectif

Cependant, l'onde s'atténue comme expliqué au paragraphe 2.2.3. Cette compensation peut être effectuée par un amplificateur non linéaire du type logarithmique en utilisant l'expression de l'atténuation (2-14). Ce type de circuit à gain variable, peu courant, peut être trouvé sur le marché à des prix relativement élevés (plusieurs dizaines de dollars). Il est possible d'utiliser une amplification par pallier comme celle dans le kit Polaroid. Il n'est pas

pratique d'utiliser les circuits de Polaroid puisque la génération de l'enveloppe est intégrée dans le même boîtier. L'amplificateur variable va être commandé séquentiellement par le micro-contrôleur. On choisit de développer notre propre circuit à gain variable à l'aide d'interrupteurs analogiques et un réseau R/2R pour l'amplification variable. L'équation du gain variable est la suivante avec a_i les positions des interrupteurs (si $a_i = 1$, interrupteur sélectionne le signal reçu, si $a_i = 0$, l'interrupteur sélectionne la masse):

$$V_{\text{sortie}} = -V_{\text{recu}} * (a_0 + 2*a_1 + 4*a_2 + 8*a_3 + 16*a_4 + 32*a_5) * 16/64$$

Tableau 4-1 : Combinaison des commandes des interrupteurs versus le gain

gain	a0	a1	a2	a3	a4	a5
0.25	1	0	0	0	0	0
0.5	0	1	0	0	0	0
1	0	0	1	0	0	0
1.5	0	1	1	0	0	0
3	0	0	1	1	0	0
4	0	0	0	0	1	0
5	0	0	1	0	1	0
6	0	0	0	1	1	0
7	0	0	1	1	1	0
8.5	0	1	0	0	0	1
10	0	0	0	1	0	1
11.75	1	1	1	1	0	1

Le séquenceur du micro-contrôleur déclenche au temps voulu les combinaisons de la commande des interrupteurs, les temps sont donnés dans le tableau 2-3. Ces temps proposés par Polaroid correspondent à un échantillonnage de la courbe d'atténuation.

La dernière étape de l'unité conversion est le convertisseur A/N. Dans de nombreux micro-

contrôleurs, ce convertisseur est intégré, mais souvent la conversion est lente. Ce facteur est une condition de sélection du micro-contrôleur. Dans le paragraphe 2.1.2, l'influence de la période d'acquisition est évaluée. La zone retenue est autour de 0.75 fois la période du signal transmis tout en l'excluant, voir 3.1.2.3. Le rapport est déterminé par les temps d'exécution du micro-contrôleur.

4.2 Étude du bloc de reconstitution

4.2.1 Principe

Le bloc de reconstitution est un bloc entièrement numérique, il sera chargé d'envoyer le signal d'excitation avec les 15 impulsions (unité contrôle) et d'effectuer l'intercorrélation entre la fonction porte du signal d'émission et la valeur absolue du signal reçu afin de trouver le maximum de la fonction d'intercorrélation qui correspondra au temps de vol. Toutes ces opérations demandent un parfait synchronisme.

4.2.2 Cahier des charges pour une structure matérielle

Tel que cela ayant été indiqué précédemment, le type de la structure matérielle est un micro-contrôleur. Ce dernier présente l'avantage d'être bien adapté à notre application, il possède des séquenceurs, des entrées et des sorties numériques, un module de conversion A/N. Maintenant, certains types de DSP incluent les fonctions du micro-contrôleur, mais la puissance de calcul de l'algorithme de mesure de distance demande une très petite ressource du DSP.

Le micro-contrôleur à choisir doit répondre au cahier des charges suivant:

1. Conversion A/N en moins de 10 μ s.
2. Compteur ayant une résolution d'au moins 5 μ s.
3. Une largeur de bus de 16 bits.
4. Une sortie servant à la génération d'un signal carré.

5. Une mémoire RAM interne suffisante pour contenir les points de mesure et ceux de l'intercorrélation (minimum de 200 octets).
6. Une mémoire ROM interne suffisante pour contenir le programme.

Ces choix se justifient de la façon suivante:

- Le séquenceur devra avoir un cycle correspondant au maximum au plus petit multiple commun entre la $\frac{1}{2}$ période d'émission (signal carré) et la période d'acquisition des données pour l'intercorrélation. Ce qui correspond à environ $\frac{1}{4}$ de la période d'émission, soit pour une fréquence d'émission de 50 kHz un cycle du séquenceur d'au plus 5 μ s.
- L'utilisation d'un micro-contrôleur rapide, ayant un bus de donnée de 16 bits, est imposée par la mesure du temps de vol qui est relativement lente et le séquenceur devra avoir cette dimension. Le transducteur a une portée de 10 mètres ce qui correspond à un temps de vol de 58 ms. $2^8 < \text{Tof} / \text{résolution} < 2^{16} \Rightarrow 16 \text{ bits}$
- Il est souhaitable d'utiliser le maximum des possibilités du micro-contrôleur, de façon à avoir le moins d'éléments sur la carte, on pourra intégrer le bloc de contrôle et une partie du bloc de conversion. Ces parties font appel à une sortie numérique sur laquelle on peut créer une onde carrée.
- Les points de l'intercorrélation et ceux de la conversion devront tenir sur la mémoire RAM. En effectuant une acquisition du résultat sur 2 fois la longueur émise, il faudra au moins une mémoire de cinq fois la longueur émise. La longueur du signal émis est de 20 points, c'est à dire 15 impulsions rectangulaires pour l'excitation. En ajoutant les variables locales de calcul, il faut une mémoire minimum de 200 octets pour la zone des variables. Le code programme va être mis en RAM (version de développement) ou en ROM (version finale) et on peut lui réserver environ 400 octets.

4.2.3 Choix de la structure matérielle

Dans un souci financier, les micro-contrôleurs sélectionnés répondent aux exigences du cahier des charges, sans avoir trop de modules qui ne servent pas ou qui sont extrêmement rapides.

Dans un souci de disponibilité et en tenant compte du nombre de fabricants de micro-contrôleurs 16 bits avec convertisseur A/N intégré sur le marché (Hitachi, Mitsubishi, Nec, Oki, Intel, SGS-Thomson, National Semiconductor, Siemens, Motorola ...), les micro-contrôleurs présélectionnés seront des marques suivantes: Intel et Motorola.

4.2.3.1 Motorola 68HC12 et 68HC16

La famille Motorola 68HC12 est l'une des plus récentes en micro-contrôleurs. Elle a été commercialisée en mai 1996. Elle présente la caractéristique intéressante d'être entièrement compatible avec le 68HC11. Son horloge peut aller jusqu'à 16 MHz, le micro-contrôleur contient un kilo octets de mémoire interne SRAM et de la mémoire morte EEPROM (4 ko pour le 68HC812A4, 32 ko pour le 68HC912B23). Il possède des compteurs 16 bits, sa conversion A/N 8 bits peut être effectuée en 10 μ s et plus, des sorties PWM qui génèrent un signal rectangulaire à rapport cyclique programmable et dispose d'un bloc de traitement des interruptions pouvant créer des événements périodiques. Ces caractéristiques sont compatibles avec l'application voulue. À cause de sa nouveauté, peu de revendeurs le proposent sur leur catalogue et aucune carte de développement n'est disponible (été 97).

La famille Motorola 68HC16 propose trois types de micro-contrôleurs 16 bits. Le type 68HCx16Xx, ne contient pas d'unité TPU pour la génération d'événements internes. Les types 68HCx16Yx et 68HCx16Zx proposent ce mode. Le 68HC16 possède les qualités voulues pour l'application et dispose de nombreux périphériques en interne. Il intègre quelques fonctions de type DSP, comme une addition-multiplication-accumulation qui se répète n fois, avec n une valeur donnée par le programmeur. Il contient au moins 1 ko de mémoire programme (mémoire morte) et un autre kilo (mémoire vive) pour les variables. Le type 68HCx16Zx propose des fréquences d'horloge allant jusqu'à 25.17 MHz, contrairement aux deux autres familles pouvant aller jusqu'à 16.78 MHz. Les petites cartes de développement sont confiées à la société Intec Automation Inc. sous le nom Freedom16. La version utilisant le 68HCx16Y1 n'est pas encore commercialisée (en été 97) et elle

propose l'utilisation des 16 canaux du "Time Processing Unit" pour générer des événements internes. Les kits de développement proposés par Motorola sont destinés à l'industrie, donc avec un prix élevé.

4.2.3.2 La famille 8xC196 d'Intel

Cette famille de micro-contrôleurs est très grande comparée au deux précédentes plus de 25 types différents. Intel les a classés en quatre types : ceux à faibles coûts; les standards; ceux dédiés au contrôle de moteurs; et ceux à hautes performances avec structure entrée/sortie avancée.

Les micro-contrôleurs à faible coût ont une horloge allant jusqu'à 12MHz, la vitesse de conversion espérée ne peut pas être atteinte. Les micro-contrôleurs standards proposent une vitesse allant jusqu'à 20 MHz, avec cette vitesse les contraintes temporelles sont résolues. Les micro-contrôleurs dédiés au contrôle des moteurs ont une fréquence de 16 MHz et possèdent plus d'entrées et de sorties, y compris le nombre de canaux de conversion passant de 8 à 12. Ce type d'entrées/sorties supplémentaires ne m'est pas utile, donc ce type ne sera pas retenu. Le dernier type ne sera utilisé que très partiellement par l'application, il permet d'atteindre des fréquences de 50 MHz.

Le type de micro-contrôleur qui convient est le type standard ou HSIO (entrées/sorties hautes vitesses) qui comporte trois séries de micro-contrôleurs, KB, KC et KD. La différence se trouve au niveau de la mémoire, et les deux derniers sont disponibles en 20 MHz. Intel propose une petite carte d'évaluation et de développement appelée sous le nom "*Target kit*". Cependant la production de cette carte est discontinuée, mais l'université en possède quelques exemplaires. Mon choix est le 87C196KD20.

Une description complète du micro-contrôleur 8XC196KD20 est faite à l'annexe E.

4.2.3.3 Description de l'application avec le 87C196KD20

L'utilisation du micro-contrôleur permet de remplacer la carte de mesure et la carte de traitement qui affiche le résultat trouvé. Ces deux cartes proviennent du "kit de développement" de chez Polaroid. Cependant, certains éléments de la carte ne pourront pas être remplacés par le micro-contrôleur. Il s'agit du bloc de commande du transducteur ultrasonore qui sélectionne si on travaille en mode émission ou pas et le module affichage qui sera optionnel, car le résultat pourra être stocké en mémoire.

Cette application demande les entrées/sorties suivantes :

- Une entrée numérique pour commander le programme
- Une entrée analogique pour l'acquisition de l'écho
- Une sortie numérique pour l'émission du signal d'excitation.
- Un port 8 bits en sortie pour la commande du gain variable.

Ces différents signaux seront traités respectivement par les périphériques internes suivants, figure 4-5 :

- Le module HSI (entrée à haute vitesse)
- Le module de conversion Analogique Numérique (ACH)
- Le module HSO (sortie à haute vitesse)
- Le Port 1 bidirectionnel

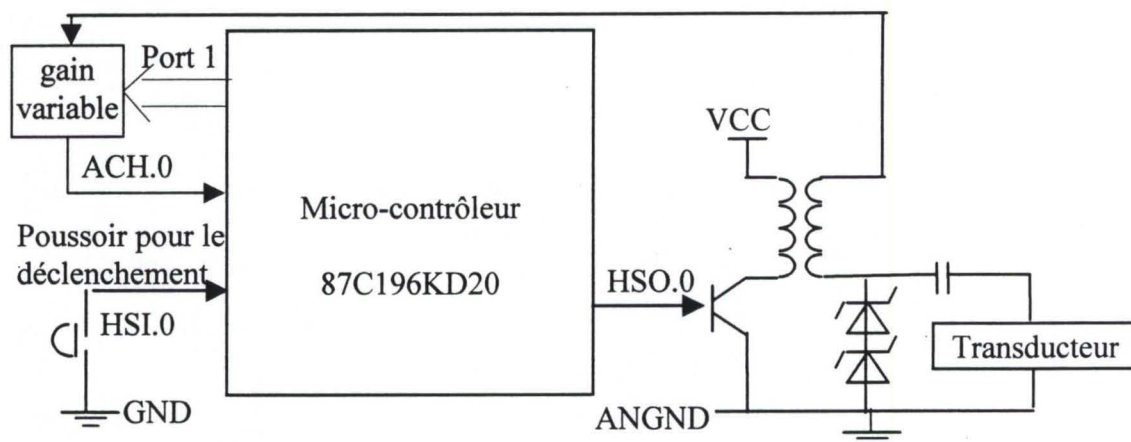


Figure 4-5 : Interface simplifiée du 87C196KD

Par soucis de ne pas occuper un grand nombre de pages du rapport, la description des fonctions qui sont spécifiques à ce micro-contrôleur suit.

4.2.3.4 Fonctionnement des événements déclenchés à l'aide du HSO

Le CAM (*Content-Addressable Memory*) est la pièce principale du module de HSO. Ce fichier emmagasine jusqu'à huit ordres. La largeur de chaque entrée du CAM est de 24 bits. Seize bits sont chargés depuis le registre HSO_TIME pour spécifier le temps d'action et huit bits sont chargés depuis le registre HSO_COMMAND pour spécifier la nature de l'action (une interruption, choix du *Timer* de référence,...).

Le HSO compare toutes les huit entrées de CAM avec la valeur de séquenceur avant de déclencher un événement. Le module de sortie à haute vitesse (HSO) déclenche des événements aux temps spécifiés en utilisant le *Timer1* ou le *Timer2* comme référence de temps. Ces événements programmables incluent le début d'une conversion A/N, l'initialisation du *Timer2*, la génération de délai à l'aide de quatre compteurs logiciels, et la mise à 1 ou à 0 des sorties du HSO (HSO.0-HSO.5). Le schéma structurel de cet organe se trouve à la figure 4-6.

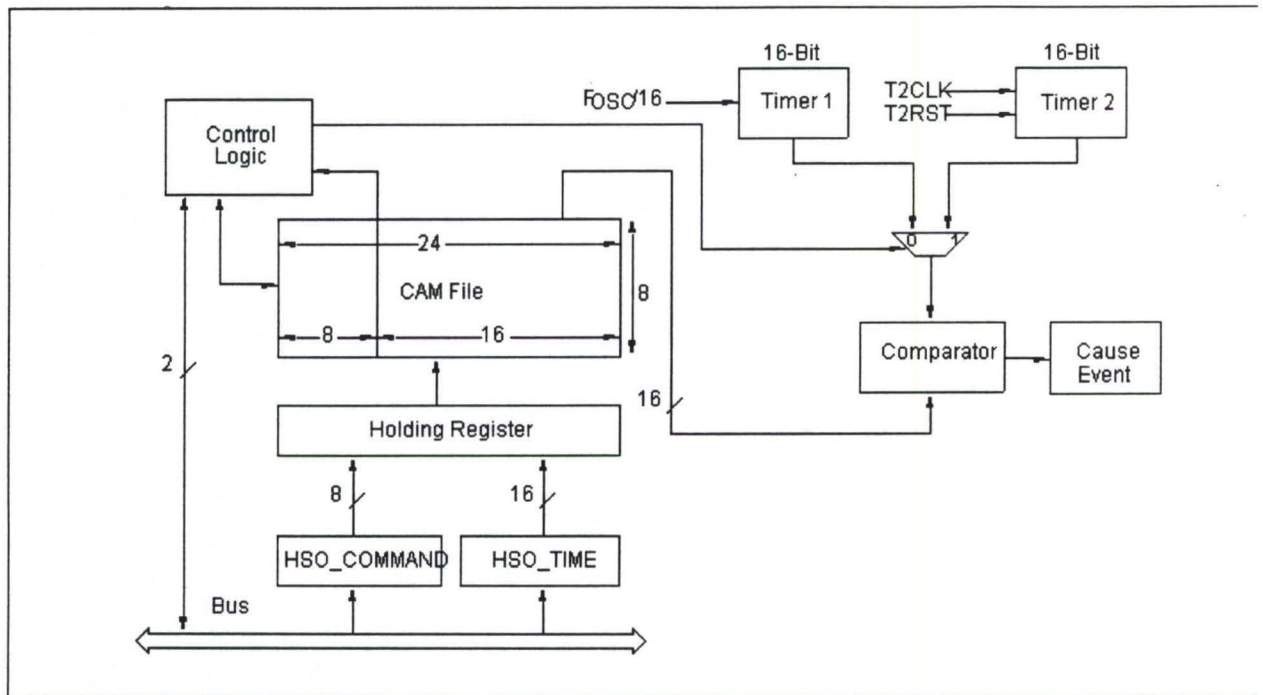


Figure 4-6 : Schéma bloc du HSO [Intel92]

Le HSO peut déclencher 15 événements différents qui sont classifiés comme externes ou internes. Les événements externes sont ceux qui basculent une ou des sorties de HSO; les événements internes sont ceux qui effectuent les tâches suivantes: démarrer des compteurs logiciels, faire un mise à 0 du *Timer2* et commencer une conversion A/N. Tous ces événements règlent des drapeaux qui peuvent engendrer optionnellement des interruptions.

Le CAM peut être aussi programmé pour garder en mémoire les événements après l'exécution ou les effacer dès qu'ils sont exécutés (registre HSO_COMMAND et IOC2). Le registre IOC2 peut être configuré de façon à faire une remise à zéro de la mémoire du CAM.

Côté physique, les sorties externes du HSO sont des collecteurs ouverts, ce qui signifie qu'ils présentent soit une masse ou un point flottant. Si on désire les rendre compatibles avec les signaux TTL, il suffit de mettre des résistances de rappel reliées à une alimentation

de 5 Volts. Cependant, ce type de sortie peut délivrer seulement 5 mA.

4.2.3.5 Fonctionnement du convertisseur Analogique-Numérique (A/N)

Le micro-contrôleur dispose des entrées (canaux) analogiques (ACH0-ACH7), qui peuvent être converties sous forme d'un résultat numérique en 8 ou 10 bits. Le convertisseur peut fonctionner sous deux modes:

- Le mode compatible 8XC196KB, où les temps d'échantillonnage et de conversion sont prédéfinis, par rapport à un nombre de cycles du *Timer2* (mode normal ou rapide).
- Le mode configurable où l'utilisateur choisit le temps de conversion et d'échantillonnage en respectant les deux formules suivantes.

$$T_{sam} = (8 * SAM + 2) / F_{osc}$$

$$T_{conv} = (16 * CONV + (2 * B + 3)) / F_{osc}$$

où F_{osc} est la fréquence du processeur, B la résolution de la conversion, SAM un code 3 bits et CONV un code 5 bits.

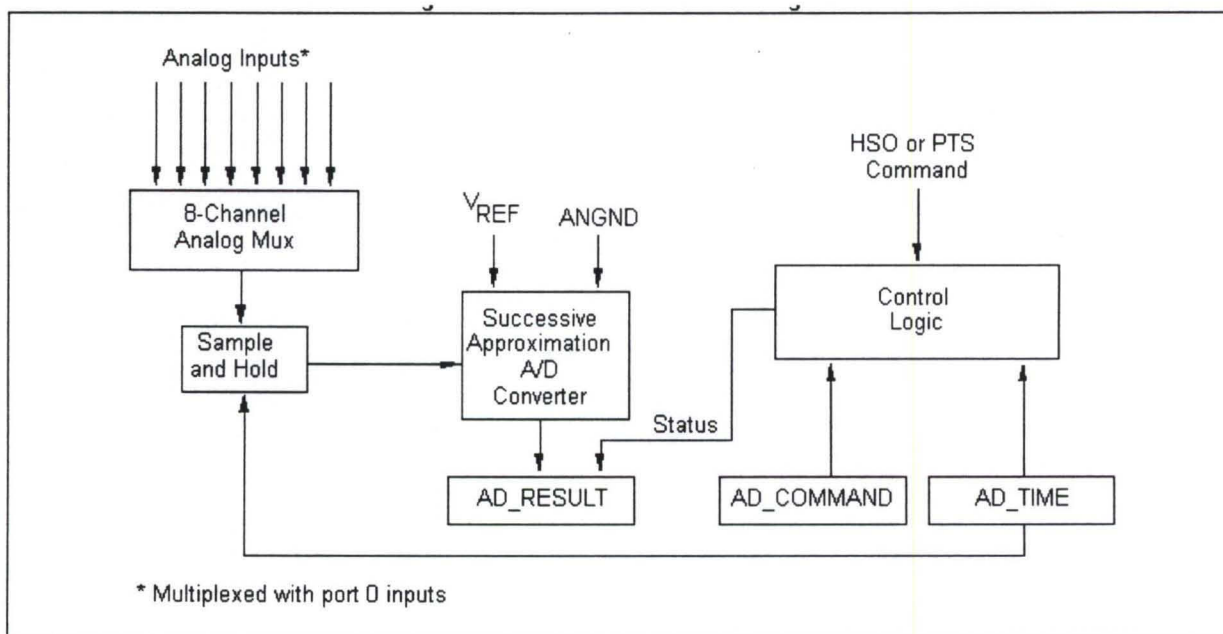


Figure 4-7 : Schéma bloc du convertisseur A/N [Intel92]

Le convertisseur peut être déclenché de plusieurs manières:

- Instantanément après la configuration.
- A un temps prédéfini (absolu ou relatif du *Timer1* ou *Timer2*), utilisation du HSO.

Ces deux modes peuvent être utilisés avec le mode *A/D Scan* qui est pré-programmé par l'utilisateur et automatisé à l'aide d'interruption de fin de conversion et du PTS, *Peripheral Transaction Server*. L'utilisateur peut aussi programmer la conversion à l'aide du HSO, avec s'il le désire des interruptions de début de conversion (INT05) et/ou de fin de conversion (INT01).

Chaque entrée du module de conversion A/N fait appel à trois entrées: il s'agit de $ANGND_i$, la masse analogique du signal à convertir; $VREF_i$, la tension maximale de l'entrée analogique; et ACH_i , l'entrée analogique à convertir qui doit être comprise entre $VREF_i$ et $ANGND_i$ (l'indice i désignant l'entrée numéro i allant de 0 à 7). Les valeurs limites sont les suivantes: $4.0 < VREF_i < 5.5$ et $V_{ss} - 0.4 < ANGND_i < V_{ss} + 0.4$

4.2.3.6 Fonctionnement des entrées à hautes vitesses (HSI)

Le module HSI permet d'enregistrer les temps des événements extérieurs avec une précision de 8 états internes ($=1$ cycle de *Timer1* $= 8 \cdot 2 / F_{osc}$).

Le module HSI contrôle quatre entrées externes. Le HSI peut être programmé pour effectuer une capture à chaque transition positive ou négative ou les deux ou tous les huit fronts positifs. Cette flexibilité permet d'utiliser le module HSI pour mesurer une variété d'entrées (largeur d'impulsion, période, rapport cyclique, différence de phase...).

Ce module peut générer une interruption si celle-ci est demandée.

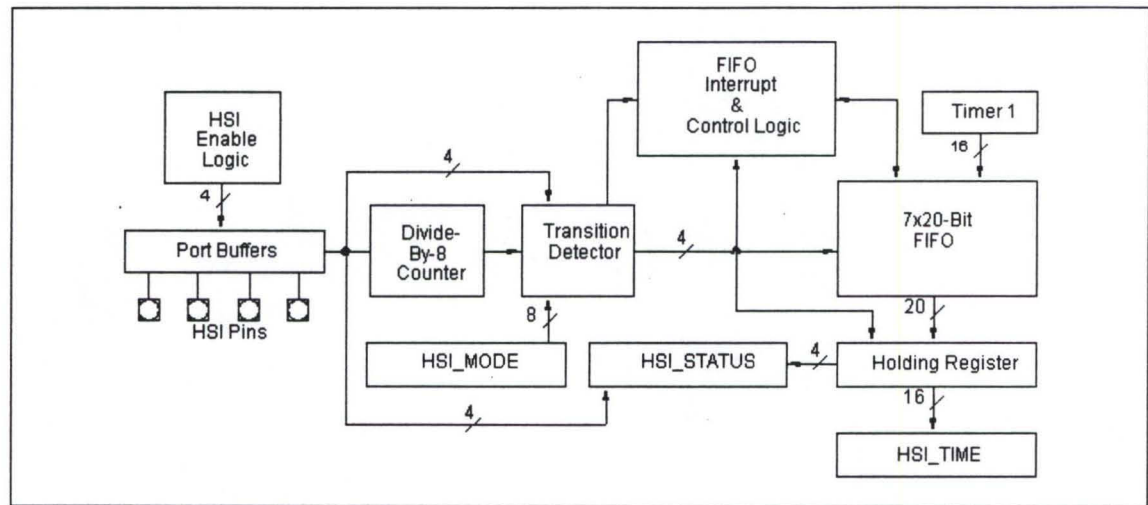


Figure 4-8 : Schéma bloc du HSI [Intel92]

4.2.4 Justification des modes du 8XC196KD pour l'application

La principale contrainte est d'avoir un parfait synchronisme temporel entre la période de coupure (émission) et la période d'échantillonnage.

⇒ Cette contrainte impose que les événements soient gérés par le même séquenceur.

- Il faut avoir un échantillonneur très rapide, pour garder en mémoire la valeur du point acquis

⇒ Pour cela, on choisira le mode de conversion programmable par l'utilisateur

Mode programmable: $T_e \geq 0.55\mu s$.

Mode compatible KB: $T_e = 0,8\mu s$ (mode rapide), $T_e = 1,5\mu s$ (mode normal).

- Choix de la méthode de déclenchement de la conversion.

⇒ On choisira le mode déclenché par le HSO

Le mode programmable de conversion est contrôlé par le HSO pour le déclenchement. Ce dernier fait appel au *Timer1* comme référence temporelle. Contrairement, dans le mode "instantané", la conversion démarre le cycle après la mise à jour du registre de commande du module.

⇒ On n'utilisera pas le mode *A/D Scan*, car ce dernier n'offre pas toute la flexibilité du HSO, malgré sa puissance. Le temps où les échantillons du signal reçu arrivent, n'est pas connu à l'avance.

- On recherche une bonne précision du résultat

⇒ Pour la simplicité du programme et de son exécution, on utilisera un résultat de conversion sur 8 bits. Par contre, la précision demande un temps de conversion le plus long possible, à cause de la méthode de conversion par approximations successives. Cependant, la somme des deux temps du convertisseur, celui de l'échantillonnage (T_{sam}) et de la conversion (T_{conv}), doit être inférieure à la période d'acquisition.

$T_{SAM}=1.05\mu s$ (SAM=3)

$T_{CONV}=8.95\mu s$ (CONV=11)

- Les autres modules peuvent-ils supporter le *Timer1* comme référence?

⇒ La sortie HSO peut être utilisée soit avec le *Timer1* ou *Timer2* en référence, l'entrée HSI utilise uniquement le *Timer1* comme référence, donc il n'y a pas de conflit. Cependant le *Timer1* ne possède pas l'événement remise à zéro contrairement au *Timer2*, mais ce point peut être contourné en changeant le mode de programmation interne du mode *Hwindows* 0 au 15. De même, la résolution du *Timer1* est de 8 états, contrairement à 1 état (mode rapide) ou 8 états (mode normal) du *Timer2*.

- Recherche de la combinaison du nombre de cycles du *Timer1* pour répondre aux exigences de synchronisme.

⇒ La période d'émission est définie par les caractéristiques du transducteur. Elle peut être générée soit par une temporisation effectuée par des instructions et par la commande du bit de sortie, soit par le *Timer1* avec le module HSO. La première possibilité offre un temps défini à l'état près contrairement à 8 états près pour le *Timer1*. La période d'acquisition doit être supérieure à la somme des temps d'échantillonnage et de conversion du micro-contrôleur. Le rapport entre ces périodes doit donner une dispersion minimale, confère la figure 3-2.

- Comment savoir si le résultat de conversion correspond au signal utile ou pas?

⇒ Il suffit d'avoir une routine qui teste le résultat dès qu'une nouvelle valeur de conversion est disponible. On a le choix entre une automatisation de la tâche avec une interruption de fin de conversion ou un simple test du bit de l'état de la conversion. On choisira la deuxième solution à cause du temps important engendré pour le traitement de l'interruption. Une contrainte sur ce programme est qu'il doit être exécuté en moins de 24 cycles du *Timer1*, soit 192 états du processeur.

- Peut-on effectuer l'intercorrélation au fur et à mesure que les résultats du signal utile arrivent?

⇒ Oui, si le calcul d'un point du signal d'intercorrélation avec le prétraitement du résultat dure moins que la période d'acquisition. Le calcul de l'intercorrélation sera effectué à la fin, car le temps d'exécution de la routine qui effectue l'intercorrélation pour un résultat reçu dépasse la limite temporelle. La conversion est effectuée matériellement, c'est à dire que le micro-contrôleur peut effectuer d'autres tâches logicielles en parallèle. Il est important quand le micro-contrôleur arrive en fin de conversion, que le programme soit en train d'attendre à la boucle de test de fin de conversion.

4.3 Implantation de l'algorithme

Les grandes lignes de l'algorithme ont été expliquées dans le chapitre 5, on a choisi la version hors ligne pour le calcul de l'intercorrélation. Dans ce paragraphe, on va expliquer le fonctionnement séquentiel de l'algorithme dans le micro-contrôleur.

La coordination des événements va être effectuée par événements internes générant une interruption ou pas. Pour chaque interruption, il y aura un sous programme appelé routine. Le programme principal sert uniquement pour initialiser et pour l'attente des interruptions qui synchronisent les événements.

Les événements vont être tous déclenchés par des interruptions, excepté le début de conversion qui va être déclenché en interne avec le CAM, la fin de conversion et aussi le signal d'excitation.

4.3.1 Algorithme du programme principal

- Faire une remise à zéro du CAM
- Autoriser les interruptions provenant d'HSI.0 externe
- Attente d'une interruption

4.3.2 Algorithme de la routine émission déclenchée par HSI.0 externe (INT04)

- Interdiction des interruptions
- Remise à zéro du *Timer1*
- Création du signal d'excitation sur le HSO.0
- Configuration du compteur logiciel 0 qui déclenche une interruption après 0.9 ms
- Autorisation des interruptions provenant de compteur logiciel, fin INT05

4.3.3 Algorithme de la routine fin de conversion

- Interdiction des interruptions
- Si le *Timer1* est en dépassement, *overflow*
 - Avertissement lumineux
- Sinon, configuration du début de conversion A/N (déclenchement par HSO)
- Répéter la routine suivante jusqu'à que la variable $n_acq = 0$
 - Vérifier l'état de la variable *acq*
 - Si $acq = 1$ (phase de mémorisation), décrémenter n_acq
 - Si $n_acq \neq 0$
 - Mise en mémoire du résultat
 - Revenir à répéter
 - Sinon, calcul de l'intercorrélation, autorisation de l'interruption sur HSI.0, retour au programme principal, *break*
 - Sinon, test du résultat ($acq = 0$)
 - Si test positif, mise en mémoire des deux résultats précédents et celui actuel, passer en mode de mémorisation ($acq=1$)
 - Si test négatif, mettre le résultat actuel dans une mémoire circulaire (3 places), fin INT01
 - Activer (mettre à jour) le gain variable
 - Vérifier si la limite temporelle n'est pas atteinte
 - Sinon autorisation de l'interruption sur HSI.0, retour au programme principal, *break*
 - Retour de la boucle répéter

Remarque : Il faut effectuer une petite normalisation du résultat de conversion avant de faire la sommation pour l'intercorrélation pour éviter un débordement. En effet les valeurs moyennes du signal reçu (idéalisé par une sinusoïde) sont d'environ 0.65 la valeur maximale (7 bits), la sommation est effectuée sur la longueur de l'émission, c'est à dire 20

échantillons. Pour faire tenir cette sommation sur 8 bits, il faudra diviser par 6.5 le signal converti.

4.4 Structure envisagée pour l'algorithme de reconstitution de position

L'algorithme présenté dans le chapitre 3, ne demande pas des ressources matérielles importantes, Le nombre d'opérations dans le cas où l'on utilise le filtre de Kalman de 10 itérations est d'environ 500 opérations de base (addition, multiplication) pour chaque couple de distances. Le temps pour faire le traitement est de l'ordre de plusieurs millisecondes pour chaque couple de distances (ce temps dépend du temps de vol). Le DSP n'est pas obligatoire pour ce type d'algorithme, un bon microprocesseur peut faire l'affaire. Le choix dépend du nombre de bits pour représenter les données dans l'algorithme, une étude de quantification est nécessaire pour déterminer le type de la structure et le type de représentation (virgule fixe ou virgule flottante).

Chapitre 5 :

Résultats expérimentaux

5.1 Méthodologie

La méthodologie pour valider le système réalisé est de tester individuellement les différentes fonctions réalisées aussi bien matérielles que logicielles. Pour la partie logicielle, les routines seront testées une à une. L'aspect temporel qui est critique pour assurer le bon fonctionnement de la méthode, sera testé expérimentalement avec des signaux synthétiques. Une fois les tests individuels concluants, des tests globaux sont effectués.

Côté logiciel :

- Calcul de la valeur absolue.
- Calcul de l'intercorrélation.
- Recherche de la position du maximum du calcul de l'intercorrélation.

Les tests consistent à écrire dans la mémoire du micro-contrôleur, à exécuter la routine et à comparer le résultat obtenu avec celui voulu.

Côté matériel :

- Test de la fréquence d'acquisition.
- Test de la fréquence du train d'ondes carrées (excitation) et de son rapport cyclique.
- Test des performances du bloc de pré-amplification
- Performance des filtres sélectifs
- Test du gain variable
- Performance de l'interface de conversion A/N

Les trois premiers tests sont spécifiques à l'application et les trois derniers seront testés à l'aide d'un écho synthétique. Le premier consiste à vérifier le temps de conversion par l'acquisition d'un signal périodique, une sinusoïde, à la fréquence voulue. Si la fréquence programmée correspond à la fréquence souhaitée, la phase des points échantillonnés sera constante. Pour le deuxième, le test vérifie si la fréquence du train d'ondes émis par le transducteur est la même que celle de résonance du transducteur. L'amplificateur de courant qui va exciter le *Ranging Module*, devra s'assurer de ne pas demander trop de courant à la sortie numérique du micro-contrôleur. On utilisera un boîtier TTL qui possède les mêmes caractéristiques que le micro-contrôleur lors du troisième test.

5.2 Résultats des tests individuels

Les résultats trouvés avec les routines de calcul du micro-contrôleur sont identiques à ceux voulus.

L'ensemble des tests matériels a été concluant. Le seul problème remarqué est que le temps donné par le constructeur pour chaque instruction et celui mesuré sont différents (test sur le train d'ondes carrées). L'explication réside dans l'utilisation d'une mémoire externe lente qui est fournie avec la *Target Board*. Normalement, le micro-contrôleur effectue une écriture en mémoire interne en 40ns. La mémoire externe qui a un temps d'accès de 30 ns, est reliée au micro-contrôleur via un verrou transparent de type D (74AC373) qui prend 6 ns comme temps d'accès. Ce qui fait 42 ns (typique) au lieu des 40 ns pour effectuer une opération d'accès. La génération du train d'ondes carrées de 15 cycles doit avoir comme fréquence 49.5kHz, c'est à dire 202 états du micro-contrôleur par période, elle prendra seulement 181 états pour générer cette fréquence.

Un autre petit problème a été l'essai avec la structure R/2R non inversée d'un convertisseur N/A de chez Analog Devices, le AD7224. Sa composition interne utilise des interrupteurs qui ne sont pas bidirectionnels, ils envoient une partie négative du signal reçu sur les lignes

d'alimentation et bruitent la mesure. Une réalisation classique a été effectuée à l'aide d'interrupteurs analogiques du type SPDT (ADG333A) et de réseaux de résistances.

5.3 Tests généraux

5.3.1 Détermination du temps d'acquisition le plus long

On évaluera précisément le parcours le plus long dans l'algorithme d'acquisition pour déterminer la fréquence d'acquisition. Pour l'acquisition, la fréquence souhaitée et calculée est de 60,6 kHz, mais à cause du temps supplémentaire pour accéder au programme (mémoire externe), la fréquence obtenue est descendue à 45 kHz. Cette fréquence est conservée puisque l'on ne se trouve pas sur un rapport entre la fréquence d'émission et celle d'acquisition multiple de 0,5. Ce rapport est de 1,1089. Si la variable HSOTIM qui représente le temps entre deux acquisitions, est sous évaluée, elle peut bloquer la phase d'acquisition et libérer le micro-contrôleur une fois que le chronomètre *Timer1* aura fait un tour complet (= 55 ms).

La valeur de HSOTIM pour les simulations est 28 cycles du *Timer1*.

5.3.2 Détermination de la résolution

La résolution temporelle de l'algorithme est donnée à la période d'acquisition près. Cette période peut s'exprimer en distance en appliquant l'équation triviale 2-28. La résolution en distance est de 3,8 millimètres pour une température de 23° Celcius.

5.3.3 Détermination de la fidélité du système

La fidélité est calculée à l'aide de l'écart type. On calibre le système en effectuant des séries de 10 mesures pour chaque distance. Les distances vont de 20 centimètres à 3,8 mètres par pas de 20 centimètres. Les temps de vol mesurés (en cycles du *Timer1*) sont présentés dans

le tableau 5-1. La résolution de la mesure correspond à la période d'acquisition, c'est à dire à 28 cycles du *Timer1*. L'écart type moyen est de 0,4125 fois la résolution. Une comparaison théorique peut être effectuée à l'aide du programme qui a généré les données pour le tableau 3-2. On ajuste le programme au rapport entre la fréquence d'émission et celle d'acquisition. L'écart type moyen trouvé est de 0,466 fois la résolution. Le système présente une très bonne fidélité. La faible dispersion des points de mesure permettra d'effectuer une calibration efficace et d'obtenir un système précis.

5.3.4 Détermination de la célérité

La mesure de la célérité est effectuée en prenant l'écart moyen en temps et en distance entre deux mesures consécutives. Le temps provient de la mesure du début de l'écho à l'oscilloscope. La distance entre l'objet et le transducteur est mesurée avec un mètre. La célérité est déterminée en appliquant la relation 5-1.

$$\text{Célérité} = \frac{\text{distance}}{0,5 * \text{temps}} \quad 5-1$$

Écart moyen de distance entre deux groupes de mesure : 0,2 mètre

Écart moyen de temps entre deux groupes de mesure : 1,565 ms

Célérité moyenne : 345,88 m/s

5.3.5 Calibration

La calibration de la mesure n'est pas évidente, car la mesure étalon est positionnée au millimètre près avec un mètre et que le sol n'est pas parfaitement horizontal. La calibration va avoir comme étalon la mesure du temps vol (début de l'écho) à l'aide d'un oscilloscope numérique qui se déclenche au début de l'émission.

distance (cm)	Tof#1	Tof#2	Tof#3	Tof#4	Tof#5	Tof#6	Tof#7	Tof#8	Tof#9	Tof#10	moyenne	écart- type	amplitude max.
20	1653	1649	1649	1649	1654	1653	1650	1649	1649	1650	1651	2.0138	3,00
40	3009	3009	3009	3009	3009	3009	3009	3009	3008	3009	3009	0.3162	1,40
60	4465	4465	4465	4465	4466	4465	4465	4465	4465	4465	4465	0.3162	0,90
80	5921	5921	5922	5921	5921	5921	5921	5921	5922	5921	5921	0.4216	0,70
100	7377	7376	7378	7377	7377	7376	7376	7377	7377	7378	7377	0.7379	1,05
120	8810	8832	8810	8834	8833	8810	8834	8834	8833	8834	8826	11.3353	0,70
140	10266	10261	10257	10261	10261	10234	10257	10233	10262	10257	10255	11.6185	1,30
160	11689	11689	11688	11666	11665	11665	11661	11689	11689	11688	11679	12.6794	0,95
180	13093	13093	13118	13145	13117	13093	13093	13093	13117	13092	13105	18.0874	0,80
200	14573	14573	14573	14601	14573	14573	14574	14597	14573	14574	14578	10.9057	1,20
220	16002	16001	16005	16002	15977	16001	15977	15977	16002	16001	15995	12.1312	1,00
240	17481	17457	17513	17458	17506	17484	17458	17458	17481	17458	17475	21.1776	1,65
260	18969	18969	18970	18970	18970	18970	18970	18969	18970	18970	18970	0.4830	1,10
280	20369	20369	20393	20370	20369	20425	20368	20369	20369	20370	20377	18.4358	1,30
300	21796	21768	21768	21817	21768	21769	21769	21769	21768	21769	21776	16.7760	1,65
320	23253	23281	23281	23281	23253	23253	23253	23230	23281	23253	23262	17.8665	1,45
340	24758	24709	24765	24765	24709	24709	24709	24709	24708	24709	24725	26.0640	1,50
360	26141	26189	26169	26166	26165	26166	26165	26165	26166	26165	26166	11.3827	1,05

Tableau 5-1 : Mesure du temps de vol

Tableau 5-2 : Répartition des mesures

0	+/- 4 ⁽¹⁾	+/- 32 ⁽¹⁾	au delà
82,94%	11,76%	5,29%	

⁽¹⁾ : en cycles du *Timer1*

Deux mesures de temps fournies par le micro-contrôleur sont importantes, le temps de vol ou Tof (Tableau 5-1 et 5-2) et le temps où le programme commence à garder les points en mémoire appelé tacq. On peut remarquer que la majeure partie des points (83%) sont données à plus ou moins 4 cycles du *Timer1*. Cette petite variation provient de la précision interne du HSO qui déclenche le début de conversion tous les 8 cycles du *Timer1*. Les temps qui sont décalés d'une résolution (HSOTIM=28) +/- la précision (=4) représentent seulement 12% des mesures, ce décalage dépend de la forme de l'écho qui n'est pas toujours constante (onde générée avec un circuit oscillant). On notera que 2 mesures sur l'ensemble de plus de 300 mesures ont donné un résultat erroné. Le système a été déclenché par impulsion de bruit qui peut être relié par une ouverture ou fermeture d'une vanne et par le démarrage de moteurs qui avaient lieu dans la pièce voisine. Selon [Borenstein95a], la quasi-totalité des fausses mesures dues aux bruits peut être détectée et supprimée en répétant une deuxième fois la mesure, ce que j'ai constaté.

L'écart moyen entre le temps relevé (entre l'émission et le début de l'écho reçu à l'aide d'un oscilloscope numérique) et le temps de vol (Tof) trouvé par le micro-contrôleur pour une même distance est de 77,87 us avec un écart type de 18,05 us.

L'écart moyen entre le temps relevé (entre l'émission et le début de l'écho reçu à l'aide d'un oscilloscope numérique) et le temps d'acquisition (tacq) trouvé par le micro-contrôleur pour une même distance est de 15,90 us avec un écart type de 17,46 us.

L'écart moyen entre le temps relevé à l'oscilloscope et celui correspondant au début de la mémorisation des points pour l'intercorrélation correspond à une période d'acquisition.

Cette période est due au seuil de détection du micro-contrôleur (pour ce test, la variable bruit = 8). Mais, cette période n'est pas prise en compte par l'intercorrélation du fait de la mémoire circulaire qui contient les trois dernières mesures.

Pour la calibration, il faut enlever en cycles du *Timer1*, la valeur 97 du temps de vol trouvé. Ce petit biais a été remarqué lors des simulations, ce dernier est constant en regardant les écarts entre deux mesures consécutives. La valeur du biais correspond à l'écart moyen trouvé.

5.3.6 Les signaux électriques

Dans le domaine électrique, seulement quatre courbes ont été retenues et montrent le bon fonctionnement du système. Les deux premières montrent l'effet du gain variable sur l'écho. La première, figure 5-1, correspond à un objet qui se trouve près, le rôle du gain est d'atténuer l'amplitude.

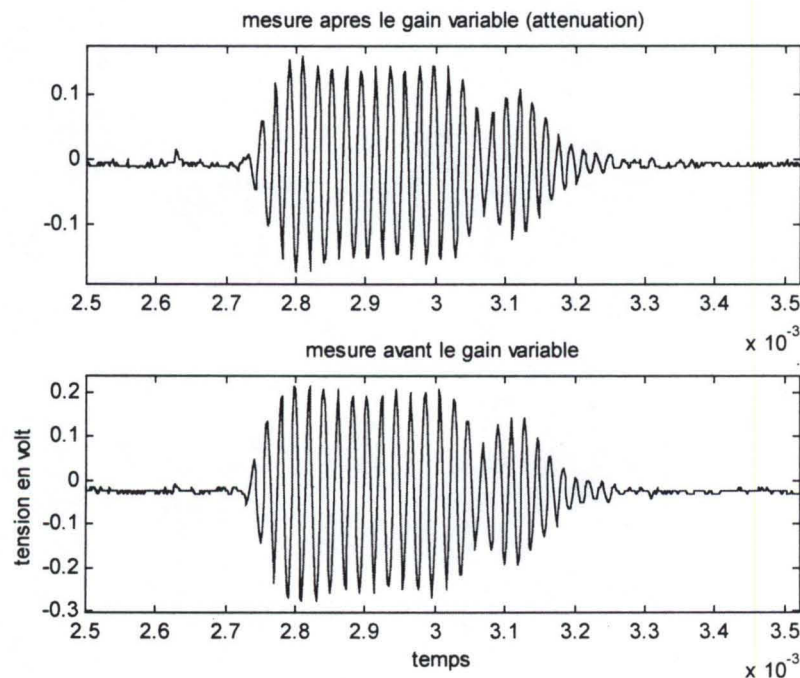


Figure 5-1 : Atténuation de l'écho avec le gain variable

La deuxième, figure 5-2, montre l'amplification de l'écho qui correspond à un objet de moyenne distance.

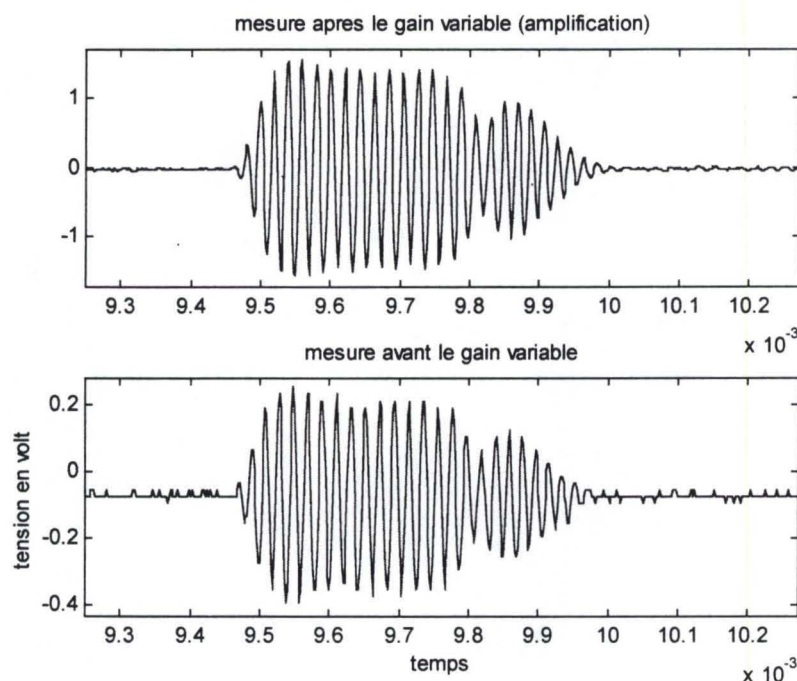


Figure 5-2 : Amplification de l'écho avec le gain variable

Dans les deux cas, le conditionnement de la mesure n'apporte pas de distorsions apparentes. Le filtrage, l'amplification, le gain variable et l'interface pour le convertisseur A/N sont correctement réalisés. Un point important est de s'assurer qu'il n'y a aucun résidu de signal qui puisse atteindre le micro-contrôleur lorsque le gain variable est désélectionné. Cette non-sélection ouvre la chaîne de mesure à l'aide des interrupteurs du gain variable. Ces derniers présentent une résistance à l'ouverture qui n'est pas infinie.

La figure 5-3 montre l'écho d'un objet à 19 centimètres. Ce chronogramme montre que le masquage de l'écho est bien fait. Le temps du premier gain variable non nul est 1,11 ms contrairement au 0,88 ms de la figure et pourrait être avancé un peu puisque sur la gauche de la figure on voit la fin de l'amortissement du signal d'émission. Je propose de garder cette condition de test pour effectuer un zoom, figure 5-4, sur le début de l'écho pour voir le niveau de bruit dans des mauvaises conditions.

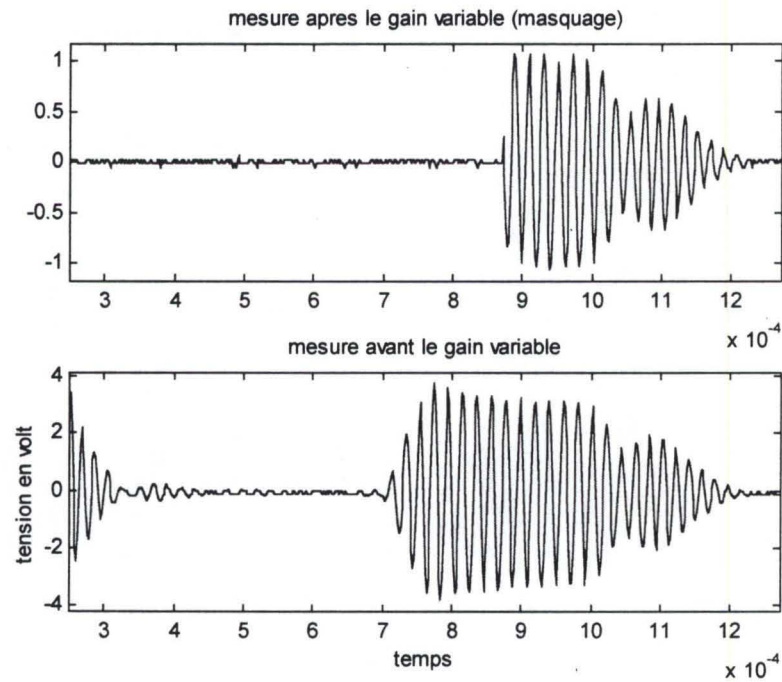


Figure 5-3 : Masquage de l'écho avec le gain variable

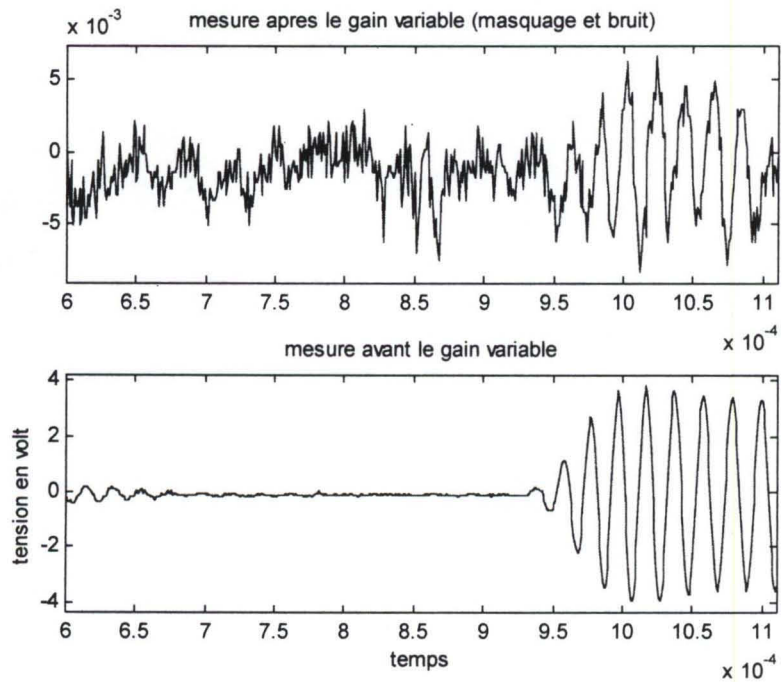


Figure 5-4 : Niveau de bruit du système d'acquisition

Le bruit de mesure est inférieur à 20 millivolts, ce qui est très bon puisque que l'on se situe sous la résolution du convertisseur 8 bits (la plus petite tension mesurée = plage de mesure sur la résolution = $5/256 = 19,5\text{mV}$ crête).

5.3.7 Action de l'alignement et des très faibles mesures

Un test supplémentaire est effectué pour voir l'amplitude minimale du signal pour que la mesure de temps de vol s'effectue correctement. Dans le programme, la variable bruit qui représente l'amplitude limite entre l'écho et le bruit de mesure, est mis à 1 (équivalent à un seuil de détection de $19,5\text{mV}$). Pour obtenir une faible amplitude de l'écho, une barre verticale de diamètre de 2 cm est positionnée dans l'axe du transducteur. Ce dernier sera décalé latéralement de façon à mesurer la variation de l'amplitude en fonction de l'angle et d'obtenir un très faible écho, le schéma de test est présentée à la figure 5-5. Une figure montrant la variation d'amplitude normalisée est montrée sur la figure 5-6, elle provient de l'amplitude la plus élevée pour une série de 10 mesures dans une même position. Le tableau 5-3 donne les statistiques des mesures effectuées par pas de 2 centimètres de variation latérale.

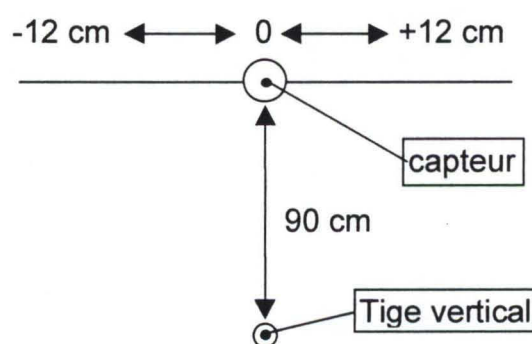


Figure 5-5 : Schéma de test

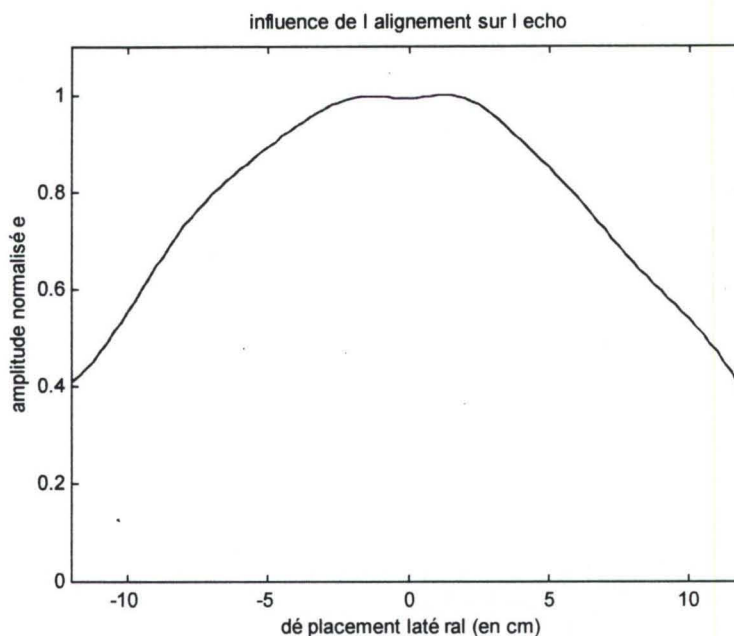


Figure 5-6 : Atténuation versus l'alignement

Tableau 5-3 : Influence de l'alignement

position (cm)	-12	-10	-8	-6	-4	-2	0	2	4	6	8	10	12
amplitude maximum (mV)	70	95	125	145	160	170	170	170	155	135	112	92	65
écart-type (mV)	36,1	0,53	0,98	0,36	0,43	0,16	0,53	0,43	0,26	0,68	0,45	1,09	92,4

La courbe présentée à la figure 5-6 présente la variation de l'amplitude en fonction du déplacement latéral. Cette courbe a été lissée par une fonction spline de Matlab et normalisée. Cette courbe présente une symétrie centrale à cause de l'alignement de l'objet avec le transducteur. Cette atténuation due aux mauvais alignements n'est pas dramatique pour la mesure du temps de vol, car l'algorithme recherche la position du maximum mais pas l'amplitude du maximum.

Cependant, cette position est bien trouvée autant que les variations des points sont suffisantes (tableau 5-2). C'est à dire autant qu'il y a suffisamment de variation du résultat quantifié. L'amplitude quantifiée unitaire représente une amplitude de 19,5mV. Pour un

écart de position de 12 cm, l'écart type trouvé est très grand ce qui donne une mauvaise mesure. Il faut un niveau crête de signal supérieur à 90mV, c'est à dire à une amplitude quantifié de 5 et plus. On remarque la performance de l'algorithme à estimer la position du maximum même avec des faibles niveaux de mesure (dès 3% de l'étendue de conversion).

Cette limite de fonctionnement doit être évitée car elle peut entraîner de fausses mesures. Une mesure simple est de mettre la variable bruit à 5. Cette solution va limiter le champ de détection du module sensoriel et va permettre de détecter un objet plus volumineux et plus lointain grâce au gain variable. L'augmentation du gain pourra remédier au rétrécissement du champ de vision, mais le bruit de mesure va croître proportionnellement, ce qui est déjà fait automatiquement avec le gain variable.

5.4 Conclusion

En conclusion, le module de mesure de distance est fonctionnel. Il permet de mesurer une distance au tiers de millimètre près à une température ambiante. La robustesse de l'algorithme a été montrée par les tests en milieux hostiles (sources de bruits ultrasonores : moteurs proches; hacheur de la climatisation...). Il s'adapte bien à la grande majorité des amplitudes de l'écho.

Chapitre 6 :

Proposition d'un système multi-capteurs

Dans le chapitre précédent, les performances du module de mesure de distance développé dans le cadre du mémoire sont analysées. La distance trouvée est donnée avec une bonne précision et une faible dispersion. Cependant cette mesure ne permet pas de positionner l'objet détecté, il faut appliquer l'algorithme de fusion présenté au chapitre 3. Dans ce chapitre, on va proposer une structure qui permettra de couvrir l'ensemble de l'environnement proche du système mobile et d'élaborer une stratégie de mesure simple.

6.1 Définition de l'application

Pour les systèmes mobiles, le principal facteur pénalisant de ces machines est la vitesse de déplacement dans des environnements partiellement connus ou inconnus. On recherche une structure qui permettra d'optimiser la vitesse de mesure, par exemple, en déclenchant plusieurs paires de capteurs à la fois s'il n'y a pas d'interférence au niveau des mesures. Dans l'algorithme de fusion qui estimera les coefficients de la tangente de l'objet détecté, il est nécessaire que les deux capteurs formant une paire, émettent en même temps. Cette structure devra être relativement simple et économique.

Dans un système mobile, en robotique, on retrouve plusieurs types de configuration de capteurs selon le type d'information voulue. Dans le cas où les capteurs ultrasonores seraient complémentaires comme avec un laser ou une caméra [Song96], quelques capteurs (1à 3) sont nécessaires. Dans les systèmes où les ultrasons sont la principale source de

mesure, on trouve un nombre de modules variant de 8 jusqu'à une couronne de 24. Lorsque ce nombre est faible, le système avance uniquement en marche avant dans une application spécifique, comme la surveillance d'entrepôt de conteneurs [Borenstein95b] où 6 capteurs sont placés à l'avant pour positionner le système lors des changements de rangées et un de chaque côté pour centrer la plate-forme dans l'allée. Cette configuration est retrouvée dans un robot agricole qui passe entre les rangées d'agrumes [Mandow96]. Les systèmes à couronne de capteurs utilisent entre 16 et 24 unités sensorielles selon l'angle d'ouverture [Elfes87, Diers96, Curran93, Irwing97]. Pour les transducteurs Polaroid utilisés, une unité est placée tous les 15° . Le cas maximal avec 24 unités est utilisé pour cette étude.

6.2 Définition de la structure

La structure doit offrir la possibilité d'activer plusieurs capteurs à la fois. L'information donnée par le module ultrasonore correspond soit à une réponse positive avec le temps de vol trouvé si un objet est détecté, soit à une réponse négative si aucun objet est trouvé ou le système est occupé. La taille de l'information retournée au superviseur dépend du compteur utilisé. Dans le cadre du mémoire, le micro-contrôleur code le temps de vol sur 16 bits. Le temps de vol le plus petit mesuré correspond à la fin du temps du régime oscillant du transducteur à l'émission. On peut insérer dans la zone non utilisée par le temps de vol, l'information concernant la réponse négative.

Pour commander plusieurs capteurs, le serveur doit pouvoir accéder à n'importe quels capteurs à la fois pour ordonner une émission simultanée des capteurs sélectionnés. Cet accès doit être effectué par des adresses. Cette adresse va être divisée en deux parties avec une partie destinée à la sélection d'un ou plusieurs groupes et l'autre partie pour sélectionner un ou plusieurs capteurs du module. On peut faire l'hypothèse que seulement une paire de capteur par module peut fonctionner en même temps.

Dans le cas d'une couronne, on peut utiliser 6 groupes qui couvrent un angle de 60° (figure 6-1). Normalement, chaque groupe peut avoir une paire de modules de mesure actifs à la fois sans avoir les interférences des autres capteurs d'un groupe voisin. Cependant le superviseur en cas de doute sur la validité des mesures peut sélectionner moins de groupes à la fois. Ce type de fonctionnement impose 6 bits pour adresser les groupes.

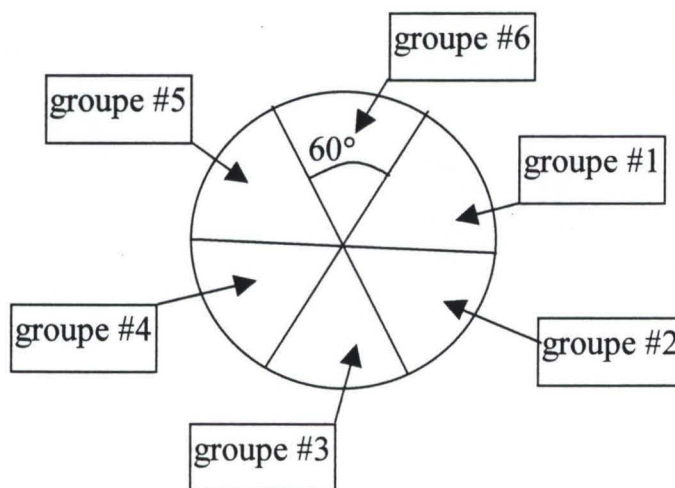


Figure 6-1 : Répartition des groupes

Pour les 4 paires de capteurs par groupe, orientées tous les 15° (figure 6-2) l'émission se fait par paire et la réception individuellement. On peut utiliser un codage de type 2 fois 2 bits, les deux premiers donnent le numéro de la paire et les deux suivants indiquent la sélection dans la paire (les deux capteurs, le capteur gauche, le capteur droit). Le bus d'adresse doit contenir 10 bits en tout (figure 6-3).

paire #1, paire #2, paire #3, paire #4

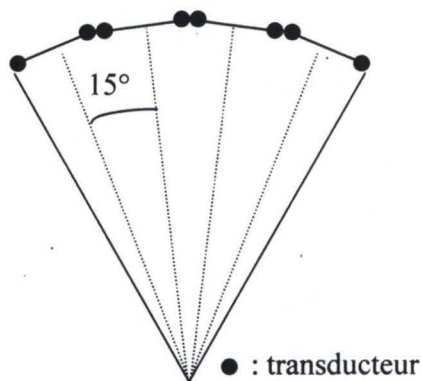


Figure 6-2 : Composition d'un groupe

Maintenant, on connaît les formats des données à communiquer entre le superviseur et le module. Il faut déterminer la façon de transférer ces données au plan physique (type de liaison avec combien de fils) et logiciel (protocole).

15															0
gr6	gr5	gr4	gr3	gr2	gr1	np1	np0	sp1	sp0	X	X	X	X	X	X
numéro du groupe						numéro de la paire		sélection de la paire		non défini					

np1	np0	# paire
0	0	0
0	1	1
1	0	2
1	1	3

sp1	sp0	sélection
0	0	interdit
0	1	a
1	0	b
1	1	a et b

Figure 6-3 : Adressage des modules

6.3 Agencement des transducteurs.

Dans chaque groupe, on retrouve 4 paires de transducteurs, les éléments de chaque paire sont notés : indice a pour ceux à gauche; b pour ceux à droite. On utilise par groupe 2 micro-contrôleurs, un par sous-ensemble d'indice a et b. Pour une paire i, le module de mesure i.a et celui i.b travailleront en parallèle. On utilisera un multiplexeur - démultiplexeur analogique 1 parmi 4 pour relier le transducteur au module de mesure. Ce type d'architecture permet d'économiser des ressources matérielles et donc de l'espace. La composition d'un demi-groupe d'indice a ou b est présentée à la figure 6-4.

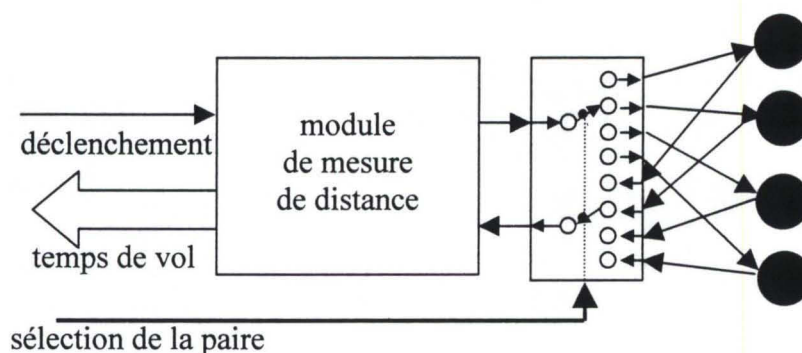


Figure 6-4 : Composition d'un demi-groupe

La distance entre les deux transducteurs de chaque paire doit être définie. Cet espacement permet de différencier un objet vu par la paire si l'objet ne forme pas un triangle isocèle dont le sommet est l'objet détecté (les deux mesures sont différentes). En réalité, la mesure de la distance qui est proportionnelle au temps de vol, est quantifiée. Avec le module de mesure conçu dans le chapitre 4, la résolution entre deux mesures est de 3,85 millimètres. Pour une même résolution, si l'espacement augmente, la différence de distance va augmenter et l'étendue de mesure commune va diminuer. Ces deux actions sont aussi importantes l'une que l'autre. La première permet à l'algorithme de fusion de donner les coefficients de droite le plus proche de la réalité. La deuxième favorisera les fausses mesures (l'objet est vu par un seul transducteur ou chaque transducteur voit un objet différent) puisque le volume où un seul transducteur émet, augmentera. Dans la littérature

[Corion96, Kuc97, Sabatini94, Song96 ...], une distance allant de 15 à 30 centimètres est proposée. La bonne résolution obtenue permet d'envisager un espacement de 20 centimètres entre les deux transducteurs d'une même paire. Cette valeur devra être ajustée expérimentalement en fonction de l'environnement.

6.4 Discussion sur le moyen de communication

On vient de proposer le placement des transducteurs avec leurs ressources matérielles qui sont partagées au sein d'un demi-groupe de 4 transducteurs. Le mode d'adressage est également défini. La donnée de mesure est envoyée en nombre de cycles du *Timer1*, elle est codée sur 16 bits. Le reste de la communication entre le calculateur et les groupes est à définir, en sachant que le calculateur peut ordonner une émission simultanée sur plusieurs groupes.

6.4.1 Type de liaison

Il existe deux grands types de liaison : la liaison série et la liaison parallèle. Pour l'application, il n'y a pas de contrainte de longueur, de rapidité qui peut imposer un type de transmission. Ici, la transmission s'effectue sur une distance maximale de quelques mètres au maximum. Pour limiter le nombre de câbles, on utilise une liaison dite d'instrumentation (figure 6-5) qui permet de brancher plusieurs appareils sur le même bus. On retrouve principalement les standards IEEE 488 pour la liaison parallèle, RS 485/422 et le nouveau standard USB pour liaison série.

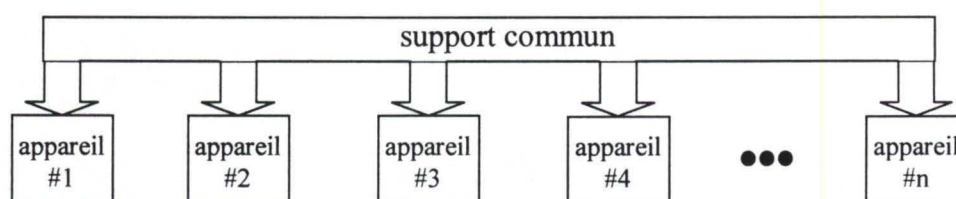


Figure 6-5 : Liaison dite d'instrumentation

6.4.2 Les interfaces séries

6.4.2.1 Les réseaux de terrain

D'autres types de câblage peuvent être utilisés comme les réseaux de terrain. Ce type de réseau utilise une interface matérielle de type RS 485 (standards Profibus et Interbus-s) ou spécifique pour les réseaux Lonworks. Les trois standards cités couvrent 75% du marché de ce type de réseau. Ces standards ont été repris par des fabricants de semi-conducteurs qui développent des circuits spécifiques, en particulier des micro-contrôleurs, comme le "Newton Chip" de chez Motorola pour les réseaux du type Lonworks. Ces réseaux sont souvent intégrés dans un système global et utilisés par l'entreprise pour commander et vérifier les différentes unités de contrôle : robots, convoyeurs,...

6.4.2.2 Les réseaux de capteurs

Dans le domaine des réseaux de capteurs, l'*Institute of Electrical and Electronics Engineers* (IEEE) avec l'implication de Hewlett Packard, *National Institute of Standards and Technology* (NIST), *Electronics Development Corporation* (EDC), ont défini un nouveau standard matériel et logiciel pour permettre d'ajouter ou d'enlever des capteurs du réseau en modifiant le moins la structure logicielle du réseau. Le protocole IEEE 1451 permet d'inclure les caractéristiques des nouveaux capteurs dans le protocole. Ces caractéristiques sont contenues dans le TEDS (*Transducers Electronics Data Sheet*) et contiennent des informations sur le fonctionnement, la plage de mesure, des informations sur la calibration ... Ce protocole définit une structure matérielle universelle comprenant 4 blocs :

1. Le NCAP (*Network Capable Application Processor*) pour l'interface du réseau.
2. Le STIM (*Smart Transducer Interface Module*) qui contient les informations sur le nombre de capteurs, la relation entre chaque capteur et le temps.

3. Le CIM (*Channel Interface Module*) qui intègre les informations physiques du transducteur, le type d'adressage, le format utilisé, la méthode de correction des données.

4. Le MMT (*Mixed-Mode Transducer*) qui comprend le transducteur et son TEDS.

Ce dernier bloc contient des données analogiques et numériques contrairement aux autres blocs qui sont entièrement numériques. Le support de communication du réseau est une liaison série. La figure 6-6 [EDC98] donne les différents points du standard par rapport aux blocs qui les relient. Ces blocs peuvent être fusionnés selon l'application.

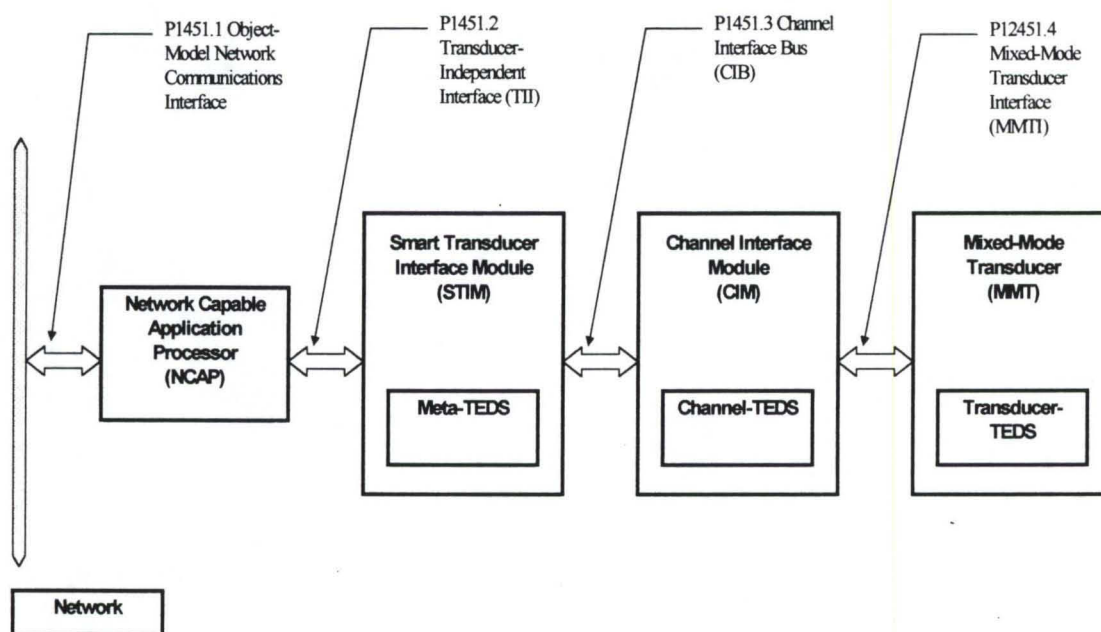


Figure 6-6 : Structure IEEE 1451 [EDC98]

Pour passer d'un type de moyen de communication à un autre, il suffit de changer le module de sortie "Network Capable Application Processor" pour qu'il comprenne le protocole du réseau. Ce protocole est en phase finale d'élaboration et présente un champ d'application très important. Dans le cas d'un système mobile de petite taille, le réseau n'est pas envisageable compte tenu de la faible distance.

6.4.2.3 Les interfaces classiques séries

Les systèmes avec USB, *Universal Serial Bus*², sont en pleine croissance. Philips, Intel, ST Microelectronics, ... proposent leur micro-contrôleur avec un USB comme interface série. Il est possible d'utiliser le protocole USB dans un micro-contrôleur ayant une simple liaison RS232 en réalisant une partie du protocole de façon logicielle et en utilisant un circuit qui assure l'interface pour les signaux. Actuellement, il n'y a pas de périphérique de communication qui intègre le protocole USB.

La liaison RS485 peut être aussi raccordée à un micro-contrôleur qui possède une liaison série RS-232, en insérant un circuit qui convertit les formats électriques et le protocole est effectué logiciellement.

Dans le cas du module de mesure réalisé, le micro-contrôleur ne doit pas être dérangé lors de la phase d'attente et d'enregistrement de l'écho car il n'a pas le temps de faire autres choses. Habituellement, un registre mémoire est inséré en interne dans le micro-contrôleur pour contenir un octet de donnée reçue. Ici, l'information envoyée et reçue est supérieure à 8 bits. Il faut prévoir des éléments extérieurs pour ne pas bloquer le bus de communication.

6.4.3 Les interfaces parallèles

Les liaisons parallèles ont comme principal standard en instrumentation, la liaison IEEE 488 ou le GPIB (*General Purpose Interface Bus*). Cette liaison permet de spécifier plusieurs fonctions aux appareils connectés comme écouteur, parleur et communicateur. Les circuits développés par les fondeurs (Texas TM9914, Motorola MC68448 ...) sont devenus désuets. Cependant la liaison parallèle peut être utilisée très facilement pour un format spécifique d'application. Le protocole peut être entièrement matériel. Des circuits intégrés commerciaux peuvent effectuer la génération de bit de parité à l'émission et

² USB est une propriété de Microsoft, Intel, Compact et NEC

vérifier ce dernier à la réception, ils possèdent un étage de sortie à trois états. L'utilisation de circuit à haute impédance permet de brancher plusieurs modules sur le même bus.

Pour rendre les données de la liaison série en données parallèles, il faut utiliser un registre à décalage en plus de la structure logique qui décode les commandes du superviseur et comprend le protocole. Cet ensemble peut être compris dans un petit micro-contrôleur ayant un bus 16 bits et une unité série. La liaison parallèle présente l'avantage d'être moins complexe puisque les échanges peuvent être gérés de façon synchrone dans un simple circuit logique. Le choix du type de la liaison va être parallèle avec ses circuits spécifiques compte tenu de la simplicité de l'interface. La structure du réseau de capteurs n'est pas entièrement définie, il manque le protocole de communication.

6.4.4 Le protocole

Il existe deux grands modes de fonctionnement, le mode maître / esclave où les esclaves répondent au maître et le mode autonome ou multi-contrôleur où chacun peut appeler et recevoir selon les règles du réseau. Dans le cadre de ce travail, le superviseur enverra les commandes aux modules de mesure et ces derniers retourneront les mesures. Ici, le temps de mesure est relativement long. Dans la configuration actuelle, le micro-contrôleur ne peut pas être dérangé lorsqu'il attend l'écho et le mémorise. Il ne sera pas utilisé pour le protocole. Une solution compacte est de donner uniquement le contrôle de la liaison au superviseur. Les groupes de mesure peuvent uniquement exécuter les commandes du superviseur et envoyer la mesure. Le superviseur doit s'assurer que le bus n'est pas inutilisable (plusieurs données présentes en même temps). Pour cela, il existe trois grands moyens de communication pour le mode autonome :

- Le moyen déterministe qui alloue une plage à chaque module pour communiquer.
- Le moyen aléatoire où tout le monde peut communiquer quand il veut, mais nécessite une vérification en cas d'émission simultanée et de reprendre un par un les envois.

- Le moyen mixte (aléatoire jusqu'à un certain nombre de communications et déterministe pour un nombre supérieur) est sans doute le plus rapide dans n'importe quel cas mais il est complexe à installer.

Les moyens déterministes sont les plus simples à gérer mais ils sont relativement lents, cette lenteur n'est pas pénalisante à cause du temps de mesure et du faible nombre de communications à effectuer. Les supports de communication du moyen déterministe se divisent en quatre groupes :

- Accès multiple à répartition en fréquence.
- Accès multiple à répartition dans le temps.
- Accès par insertion de registre.
- Méthode du jeton.

Les deux premières découpent respectivement la bande passante et le temps, de façon à allouer une plage unique à chaque communicant. L'accès par insertion de registre s'applique à une topologie en anneau où à chaque nœud, la trame se charge ou se vide des données en fonction de l'opération. La méthode du jeton correspond à un permis d'émettre qui circule pour donner ce droit au nœud. La méthode déterministe choisie est l'accès multiple à répartition dans le temps bien que la méthode du jeton soit populaire. La raison est que le temps entre deux mesures provenant du même groupe est important et quelle est simple à mettre en place, insertion d'un compteur qui active les modules à un retard prédéfini. Cette méthode aura un cycle de 13 plages temporelles (2 pour chaque groupe (module gauche et droit) plus une pour le maître). Une autre solution simple peut être choisie en utilisant une structure maître esclave qui va questionner les demi-groupes pour connaître leur état. Seul le module identifié dans la requête envoyée par le maître peut répondre. Cette méthode permet de sélectionner le répondant au lieu de tous les questionner y compris ceux qui ont déjà envoyé leur réponse. Cette solution est retenue comme proposition de câblage pour relier les transducteurs au superviseur qui sera en charge de calculer l'algorithme de fusion. Pour pouvoir effectuer une émission simultanée de plusieurs groupes à la fois, il faudra interdire la réponse des groupes. Le superviseur doit préalablement connaître l'état des

modules de mesure. On utilise le bit 5 de l'adresse (figure 6-3) pour mettre le drapeau lecture/action. Ce drapeau est à 1 en cas de lecture individuelle d'un demi-groupe et à 0 pour un ordre de déclenchement d'une nouvelle mesure. Les bits 0 à 4 de l'adresse qui ne sont pas utilisés, seront forcés à zéro pour l'émission.

6.5 Définition de la structure proposée

La description de la communication entre le maître et les esclaves (les 6 groupes de mesure) est la suivante. Il s'agit d'un protocole synchrone à temps réparti où l'horloge est distribuée par le maître aux esclaves. Chaque esclave, comme le maître, possède un compteur 3 bits qui synchronise les communications à l'aide du bloc de contrôle de l'interface avec le bus.

Lorsque le maître émet, il prend possession du bus et tous les esclaves écoutent, les esclaves lisent l'adresse émise et vérifient la parité. Le maître peut réinitialiser le cycle de communication si une erreur de parité se produit. Le message est décodé dans le bloc de contrôle et appliqué s'il concerne le groupe. Le maître peut aussi initialiser le cycle de communication si le message envoyé ne demande pas de réponse des esclaves. La deuxième moitié du cycle est consacrée à la réponse d'un groupe. Seul le groupe qui a reçu la commande d'envoyer sa donnée, émet, les autres ont leurs sorties en haute impédance. Le maître lit la réponse et vérifie la parité avant de placer le message en mémoire et de passer à l'instruction suivante. En cas d'erreur, le message n'est pas conservé et on émet à nouveau l'adresse en cours au prochain cycle.

L'architecture de l'esclave reprend le module de mesure de distance du chapitre 4 auquel on a ajouté un bit d'enregistrement qui place les 16 bits du code du micro-contrôleur dans le registre. Le cycle d'émission est toujours activé par une interruption provenant du bloc de contrôle. La nouveauté réside dans l'insertion d'un double multiplexeur - démultiplexeur analogique 1 parmi 4 qui sélectionne la paire de capteurs correspondant à l'adresse envoyée par le groupe.

La composition de l'esclave comprend les deux demi-groupes identiques (figure 6-4) et l'interface qui isole les modules de mesures du bus de communication. L'isolation des deux micro-contrôleurs est effectuée par deux registres qui conservent les réponses des 2 modules de mesure. La communication avec le bus est assurée par un circuit d'interface bidirectionnel que gère la parité. Le bloc de contrôle synchronise les circuits d'isolation selon le protocole établi à la figure 6-8. L'organisation structurelle de l'esclave est illustrée par la figure 6-7.

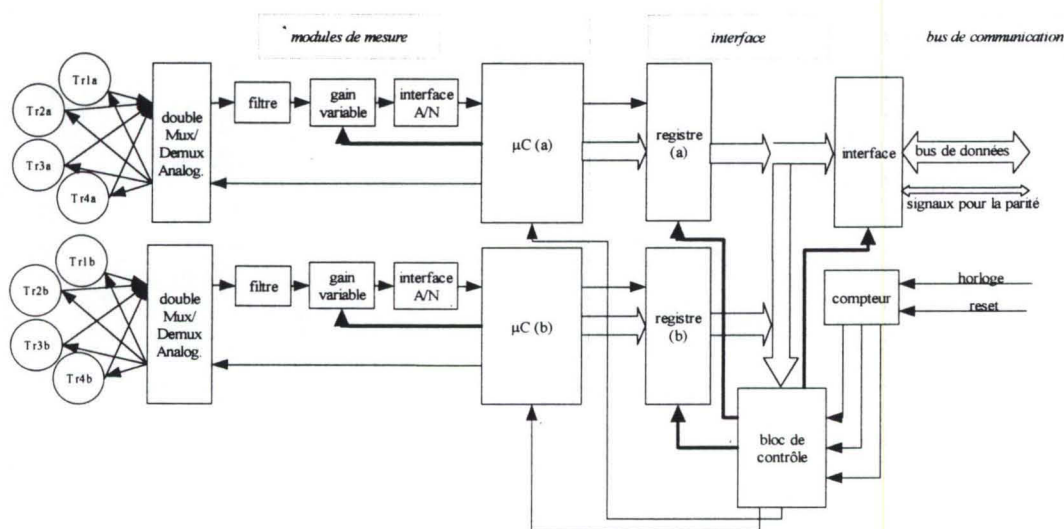


Figure 6-7 : Organisation structurelle d'un esclave

Le circuit registre peut être composé d'un réseau de 16 bascules D synchrones avec sorties à haute impédance, par exemple les circuits TTL 74xx16334 ou 74xx16646. Le circuit d'interface peut être un *16 bits universal transceivers bus with parity generators/checkers and 3-state outputs* ce qui veut dire un émetteur - récepteur 16 bits possédant des sorties hautes impédances et la génération - vérification de parité, par exemple le circuit TTL 74xx16657. Le circuit de contrôle est un circuit logique où le compteur peut être intégré.

Les figures 6-8 et 6-9 contiennent le protocole de communication vu respectivement par le maître et les esclaves.

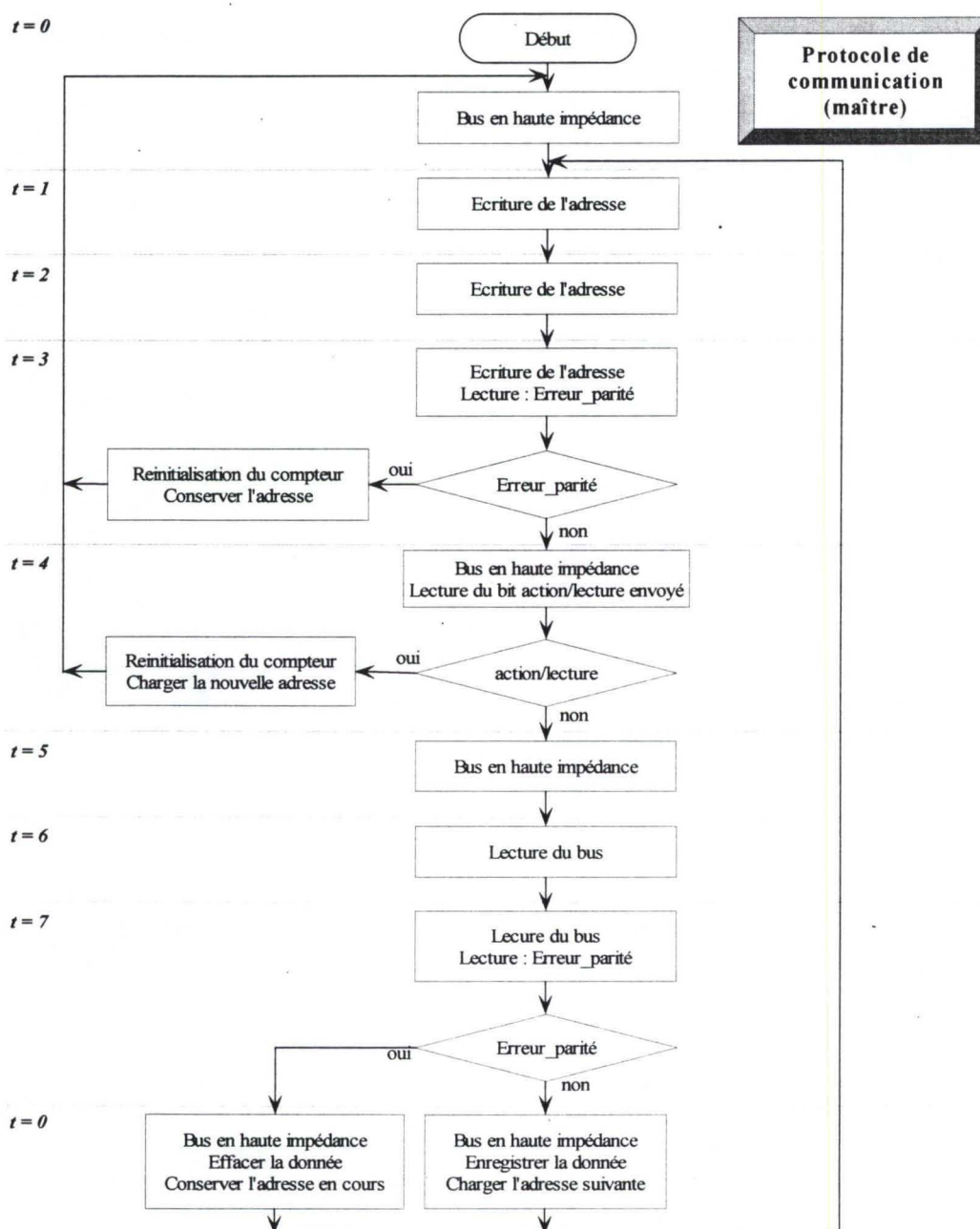


Figure 6-8 : Protocole de communication du maître

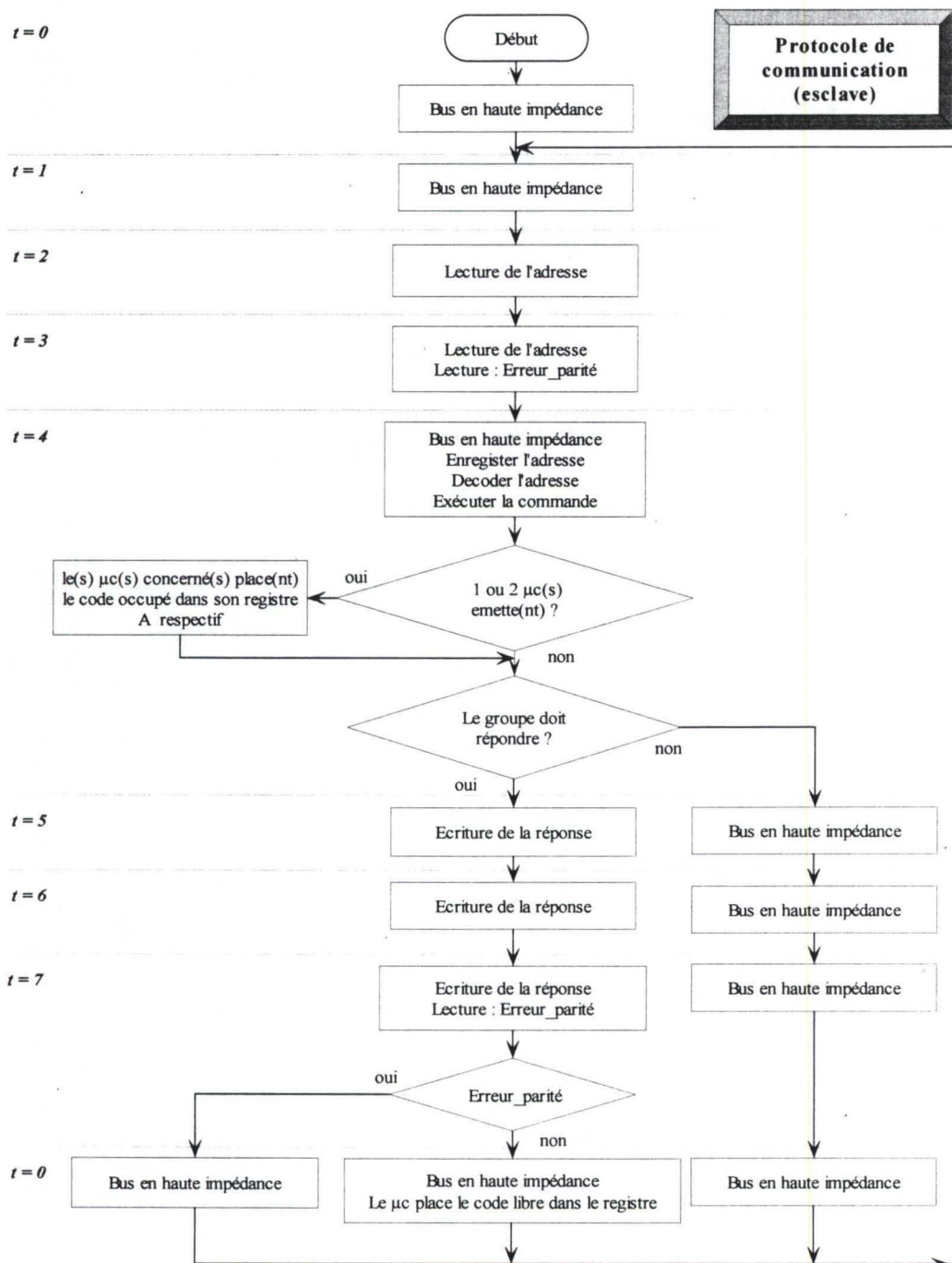


Figure 6-9 : Protocole de communication des esclaves

Sur ces organigrammes représentant le protocole de communication synchrone, il est important de placer sur le bus, la donnée qui va être lue de l'autre côté du bus avant que le module récepteur lise le contenu du bus. C'est par cette raison qu'aux instants $t = 1, 2$ pour

le maître (figure 6-8) et $t = 5,6$ pour l'esclave (figure 6-9), le bus est maintenu en écriture. De plus, le maître contrôle le bus et c'est lui qui force la réinitialisation du protocole en cas d'erreur de transmission, c'est à dire qui force à zéro tous les compteurs. Le contrôle de la parité est commun au bus de communication.

L'organisation structurelle du maître est donnée à la figure 6-10, elle reprend la même organisation des communications que celle des esclaves, les deux modules de mesure ont été remplacés par un couple micro-processeur et DSP qui est en charge respectivement de la gestion des données et du calcul des coefficients de la tangente. Une mémoire remplace les registres et est adressée par le bloc de contrôle. Une fois les coefficients calculés, ces derniers seront transférés vers un interpréteur d'environnement qui communiquera ses résultats au calculateur de trajectoire, cette partie ne fait pas l'objet du mémoire.

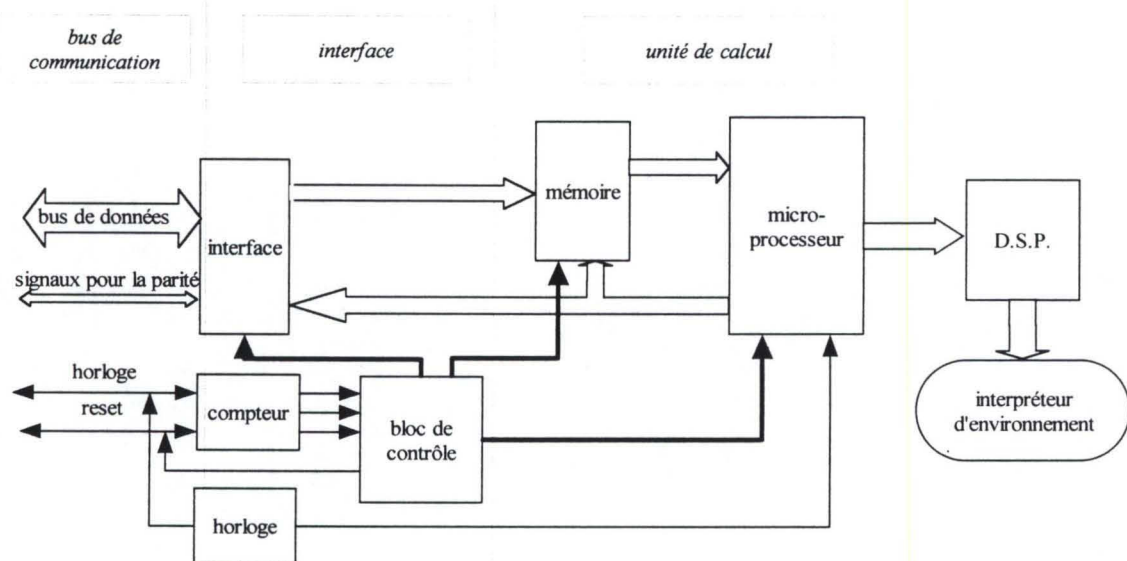


Figure 6-10 : Organisation structurelle du maître

La taille du bus est la suivante : les données à transmettre sont 16 bits auxquels on ajoute un bit de parité, un bit d'erreur de parité par octet et le signal d'horloge avec le signal de remise à zéro du compteur. En tout il y a 22 fils qui sont complétés par des fils de masse.

Les standards sont de 25 fils pour des connecteurs de type D-Sub, appelé communément DB25 et de 26 fils pour un connecteur droit ou IDC (pour des nappes de fils).

L'interface proposée pour relier les différents modules de mesure au superviseur est optimisée pour l'application. Son protocole synchrone est réalisé matériellement. Cette architecture compacte peut facilement être adaptée à différentes configurations de capteurs.

Chapitre 7 :

Conclusion

Dans les chapitres précédents nous avons vu successivement une présentation des caractéristiques ultrasonores, le développement d'algorithmes de mesure et de fusion, la réalisation d'un module de mesure de distance et une proposition d'un système multi-capteurs. Nous dressons maintenant le bilan du travail effectué.

7.1 Bilan

Les objectifs fixés au début du mémoire sont atteints.

Le premier sous-objectif consistait à réaliser un module simple de mesure de distance à la fois fiable et précis, à partir d'un transducteur ultrasonore. Les résultats obtenus avec l'algorithme qui recherche le temps de vol à l'aide de la méthode de l'intercorrélation, permet d'obtenir précisément la position du début de l'écho. La série des différents tests sur le choix des paramètres de l'algorithme valide et optimise la méthode de façon à avoir une bonne fidélité, un écart type inférieure à 0,5 en pas de mesure. Les essais expérimentaux effectués avec le module de mesure de distance ont bien confirmé les performances de cette méthode, l'écart type mesuré est de 0,412.

Le deuxième sous-objectif était de développer un algorithme de fusion, estimation des coefficients de la tangente, qui soit robuste. Cet algorithme a été plus compliqué à mettre en œuvre du fait des simulations avec un niveau de bruit élevé et une résolution de mesure faible. Les équations du système pour cet algorithme doivent être bien posées et observables pour éviter des erreurs d'estimation trop importantes. L'algorithme développé

est basé sur une étude comportementale des réponses des algorithmes du type RLS. Il permet par des moyens simples (réinitialisation, facteurs d'oubli et d'immunité...) d'obtenir un estimé rapide et précis malgré les conditions de simulation. Les changements brusques d'environnement, typiques des mesures ultrasonores, ont permis de vérifier les performances de l'algorithme.

La proposition d'un système multi-sensoriel est basée sur le travail effectué : les systèmes de mesure qui sont reliés à un unique calculateur contenant l'algorithme de fusion. Cette proposition comprend le protocole de communication et les schémas structurels. La réalisation de système peut être conçue sur une plate forme mobile.

7.2 Contributions

Les contributions apportées par ce travail sont les suivantes :

- L'étude préliminaire sur les paramètres de l'algorithme de positionnement du début de l'écho permet d'obtenir une très faible dispersion des mesures.
- Le générateur de données synthétiques ultrasonores (Annexe C) est un outil simple et configurable pour simuler l'algorithme de fusion dans un environnement virtuel.
- Les méthodes de la commande adaptative (algorithme RLS_EFRA) sont appliquées à l'algorithme de fusion pour suivre un estimé qui varie brutalement.
- Le développement de l'algorithme RLS_IMCEVI offre un estimateur simple et dédié à la fusion de données ultrasonores.
- La simplicité de l'interface développée permet de relier les modules de mesure au calculateur et peut être utilisée pour d'autres arrangements sensoriels en modifiant uniquement le bloc de contrôle.

7.3 Applications et développements futurs

Les points ci-dessous donnent des pistes pour des applications futures du travail proposé

dans le cadre du mémoire.

- Adaptation automatique de la plage de conversion pour l'acquisition, mise en œuvre de régulation à boucle fermée pour le gain variable. Limiter le gain pour avoir un rapport signal sur bruit convenable.
- Développer un système de mesure avec une paire de transducteurs qui pourront générer les couples de points pour valider expérimentalement l'algorithme de fusion en vue d'en trouver les paramètres (quantification, format...).
- Réaliser le système de mesure ultrasonore avec l'algorithme de fusion embarqué.
- Intégrer les modules de mesure pour former un capteur intelligent. Regarder les possibilités du nouveau circuit de chez Analog Devices : Le *MicroConverter AD μ C812* qui possède des convertisseurs A/N et N/A rapides, trois chronomètres 16 bits, une architecture de micro-contrôleur, un capteur de température intégré... .
- Dans le cas d'un système mobile complet, utiliser un même standard pour tous les capteurs et actionneurs comme celui IEEE 1451 avec les *TEDS*.
- Le système complet peut être aussi bien utilisé sur des robots mobiles que dans une automobile (système de détection de présence, anti-collision à faible vitesse...) en ajustant la configuration des capteurs.

Bibliographie

- [Arikan95] O.Arikan, B.Barshan, "A comparison of two methods for fusing information from linear array of sonar sensor for obstacles localisation", *Proc. Int. Conf. on Intelligent Robots and Systems (IROS'95)*, 1995
- [Astrom95] K.J.Astrom, B.Wittenmark, Adaptive control, second edition, Addison-Wesley Publishing Compagny, 1995
- [Ayrulu97] B.Ayrulu, B.Barshan, S.W.Utete, "Target identification with multiple logical sonars using evidential reasoning and simple majority voting", *Proc. IEEE Int. Conf. on robotics and automation*, Avril 1997
- [Baglio96] S.Baglio, S.Graziani, N.Pitrone, "Intelligent measuring systems for distance estimation based on fuzzy data fusion", *Proc. IEEE on instrumentation and measurement technology conference (IMTC'96)*, Juin 1996
- [Barshan90] B.Barshan, R.Kuc, "Differentiating sonar reflections from corner and Planes by employing an intelligent sensor", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 12, No 6, Juin 1990
- [Barwicz96] A. Barwicz, "A system approach to electrical measurements", *Measurement*, Vol. 19, No.3, 1996
- [BenSlima93] M.Ben-Slima, Étude d'applications d'un processeur numérique de signaux (DSP) dans les systèmes de mesure électroniques, mémoire de maîtrise, UQTR, 1993

- [Bentley95] J.P.Bentley, Principe of measurement systems, 3^{ème} édition, Longman scientific & technical, 1995.
- [Biber80] C.Biber, Polaroid: ultrasonic ranging using system handbook, 1980.
- [Borenstein91] J.Borenstein, Y.Koren, "Histogrammic In-Motion Mapping for Mobile Robot Obstacle Avoidance", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, pp 278-288, Juin 1991
- [Borenstein95a] J.Borenstein, Y.Koren, "Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance", *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 1, pp 132-138, Février 1995
- [Borenstein95b] J.Borenstein, Y.Koren, "Mobile Robot Navigation in Narrow Aisles with Ultrasonic Sensors", *ANS 6th Topical Meeting on Robotics and Remote Systems*, Février 1995
- [Byte80] "Ultrasonic ranging system: Home Iron the range", *BYTE Magazine*, Novembre 1980.
- [Chong97] K.S.Chong, L.Kleeman "Sonar Feature Map Building for a Mobile Robot", *Proc. IEEE International Conference on Robotics and Automation*, 1997
- [Corion96] O.Corion, A.M.Desodt, D.Jolly , "Using ultrasonic means for the recognition of real space", *Proc. int. symposium on measurement and control in robotics (ISMRC'96)*, Mai 96.
- [Craig89] J.J.Craig, Introduction to robotics: mechanics and control, Addison-Wesley, 1989

- [Diers96] R.Diers, K.P.Holzhausen, "Ultrasonic and laser based robot navigation", *Proc. int. symposium on measurement and control in robotics (ISMRC'96)*, Mai 1996.
- [Curran93] A.Curran, K.J.Kyriakopoulos, "Sensor-based self-localization for wheeled mobile robots", *Proc. 1993 IEEE int. conf. on robotics and automation*, 1993.
- [Elfes87] A.Elfes, "Sonar-based Real-World navigation" *IEEE Transactions on Robotics and Automation*, Vol. 3, No. 3, Mai 1987.
- [Everett95] H.R.Everett, Sensors for mobile robots: Theory and application, A.K Peters Ltd, 1995.
- [Fatemi96] A.Fatemi, H.Lecocq, "Multi-sensor fusion for mobile robot", *Proc. int. symposium on measurement and control in robotics (ISMRC'96)*, Mai 1996
- [Ferraris93] F.Ferraris, U.Grimaldi, M.Parvis & al., "A neural network for fast-response ultrasonic distance sensors", *Proc. IEEE on instrumentation and measurement technology (IMTC'93)*, Mai 1993
- [Frey69] A.E.Frey, L.E.Kinsler & al., Fundamentals of acoustics, John Wiley and Sons, 1969.
- [Gueuning96] F.Gueuning, M.Varlan, C.Eugène, P.Dupuis, "Accurate distance measurement by an autonomous ultrasonic system combining time-of-flight and phase-shift methods", *Proc. IEEE on instrumentation and measurement technology conference (IMTC'96)*, Mai 1996
- [Hanebeck94] U.D.Hanebeck, G.Shmidt, "A new high performance multisonar system for fast mobile robots", *Proc. Int. Conf. on Intelligent Robots and Systems (IROS'94)*, 1994

- [Horn97], O.Horn, A.Courcelle, "Localisation statique d'un fauteuil électrique autonome", *Journal européen des systèmes automatisés (JESA)*, Vol. 31, No 5, 1995
- [Intel92] User's manual, 8XC196KC/8XC196KD, Intel, Santa-Clara CA, 1992
- [Irwin97] J.D.Irwin, L.Kleeman, The industrial electronics handbook, section 53.3, CRC Press & IEEE Press, 1997.
- [IRSST95] J.Hurteau, R.Saint-Amant, M.Côté, "Étude des systèmes de détection d'humains et d'anticollision pour des véhicules opérant en milieu minier", *Rapport de l'IRSST (R-109)*, Québec, 1995
- [Kleeman95] L.Kleeman, R.Kuc, "Mobile robot target location classification", *International journal on robotics research*, vol. 14, No. 4, Août 1995
- [Kleppe82] J.Kleppe, Engineering applications of acoustics, Artech house, 1982
- [Ko96] J.H.Ko, W.J.Kim, M.J.Chung, "A method of acoustic landmark extraction for mobile robot navigation", *IEEE transactions on robotics and automation*, Vol 12, No 3, Juin 1996.
- [Kocis96] S.Kocis, Z.Figura, Ultrasonic measurements and technologies, Chapman & Hall, 1996
- [Kuc92] R.Kuc, B.Barshan, "Bat-like sonar system for obstacle localization", *IEEE Trans. Systems, Man and Cybernetics*, Vol. 22, No. 4, 1992
- [Labarrère88] M.Labarrère, J.P.Krief, B.Gimonet, Le filtrage et ses applications, Cepadues-Editions, 1988

- [Leonard92] J.J.Leonard, H.F.Durrant-Whyte, Directed sonar sensing for mobile robot navigation, Kluwer academic publishers, 1992
- [Mandow96] A.Mandow, J.M.Gomez de Gabriel, J.L.Martinez & al., "The Autonomous Mobile Robot AURORA for Greenhouse Operation", *IEEE Robotics and Automation Magazine*, Vol. 3, No. 4, Décembre 1996
- [Maslin83] G.D Maslin, "A simple ultrasonic ranging system", Polaroid: Ultrasonic ranging system handbook, 1983.
- [Matthlies88] L.Matthies, A.Elfes, "Integration of sonar and stereo range data using a grid-based representation", *Proc. IEEE 1988 int. conf. on robotics and automation*, 1988
- [Najim88] M.Najim, Modélisation et identification en traitement du signal, Masson, 1988
- [Polaroid83] Polaroid, Ultrasonic ranging system handbook: application notes/technical papers, Polaroid 1983.
- [Peremans93] H.Peremans, K.Audenaert, J.Van Campenhout, "High-resolution sensor based on tri-aural perception", *IEEE transaction on robotics and automation*, Vol. 9, No. 1, Février 93
- [Parrilla91] M.Parrilla, J.Janaya, C.Fritsch, "Digital signal processing techniques for high accuracy ultrasonic range measurements", *IEEE transactions on instrumentation and measurement*, Vol. 40, No 4, Août 1991.
- [Pruski96], A.Pruski, Robotique mobile, la planification de trajectoire, Hermès, 1996.
- [Sabatini94] A.M.Sabatini, "Statistical estimation algorithms for ultrasonic detection of surface features", *Proc. int. conf on intelligent robots and systems (IROS'94)*, 1994

- [Salgado88] M.E.Salgado, G.C.Goodwin, R.H.Middleton, "Modified least squares algorithm incorporating exponential resetting and forgetting", *International journal control*, Vol. 47, No 2, Juin 1988
- [Schaft96] R.D.Schaft, J.Dahlkemper "Ultrasonic phased array scanner for measurement and control in robotics", *Proc. int. symposium on measurement and control in robotics (ISMCR'96)*, Mai 96.
- [Song93] K.T.Song, C.C.Chang, "Ultrasonic sensor data fusion for environment recognition", *Proc. Int. Conf. on Intelligent Robots and Systems (IROS'93)*, 1993.
- [Song96] K.T.Song, W.H.Tang, "Environment perception for a mobile robot using double ultrasonic sensor and a CCD camera", *IEEE transactions on industrial electronics*, Vol. 43, No 3, Juin 1996.
- [Webb96] P.Webb, C.Wykes, " High-resolution beam forming for ultrasonic array", *IEEE transactions on robotics and automation*, Vol. 12, No 1, Février 1996

ANNEXES

Annexe A :

Compléments ultrasonores

A.1. Modélisation du transducteur (suite chapitre2)

Dans le paragraphe 3 du chapitre 2, une modélisation du transducteur est donnée par rapport à sa structure mécanique (figure 2-13) dont son comportement est identifié avec des éléments électriques équivalents (figure 2-14). Dans cette annexe, la modélisation du transducteur va servir à décrire la relation directe et inverse sous forme de schémas électriques et d'équations.

A.1.1. Le modèle électrique en transmission

On rappelle le schéma bloc de la relation inverse (A-1)

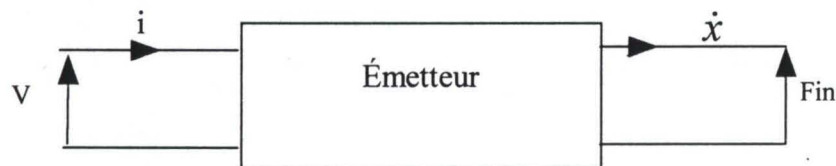


Figure A-1 : Relation inverse

Dans l'équation 2-24, $F=k_r.d_r.V$ on retrouve le couplage entre l'étage d'entrée et celui de sortie dans le schéma électrique (A-2)

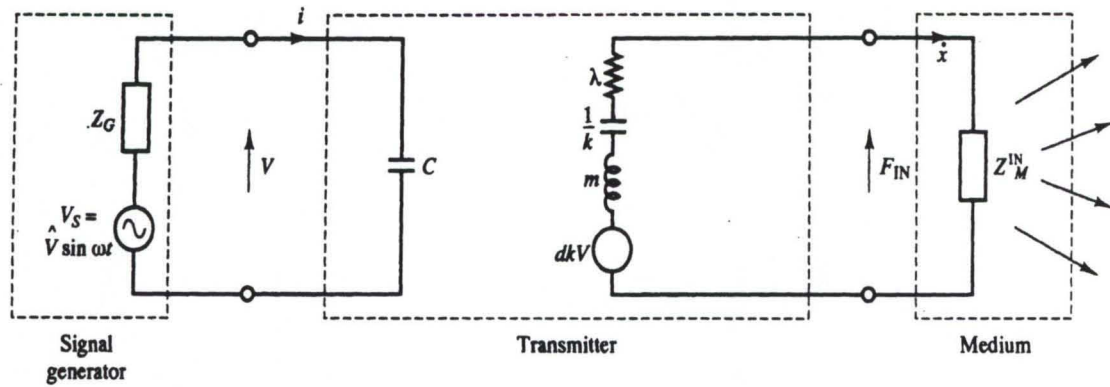


Figure A-2 : Modèle électrique en transmission [Bentley95]

Où Z_G est la résistance interne du générateur

$$V = \frac{1}{Z_G + \frac{1}{j.C.\omega}} V_s \approx V_s \text{ (avec } Z_G \text{ proche de zéro)} \quad (\text{A-1})$$

En égalant l'équation différentielle avec la relation inverse, on obtient l'équation A-2.

$$d.V.k = m.s.\dot{x} + \frac{1}{k.s}\dot{x} + \lambda.\dot{x} + Z_{IN}.\dot{x} \quad (\text{A-2})$$

$$x = \frac{1}{s} \frac{d.V.k}{m.s + \frac{1}{k.s} + \lambda + Z_{IN}} \quad (\text{A-3})$$

Comme Z_{IN} représente la résistance de l'air et celle-ci est négligeable devant les autres impédances du système, nous aurons:

$$\frac{x}{V_s} = \frac{d}{\frac{1}{\omega_n^2} s^2 + \frac{2.\xi}{\omega_n} s + 1} \quad (\text{A-4})$$

A.1.2. Le modèle électrique en réception

On rappelle le schéma bloc de la relation directe (figure A-3)

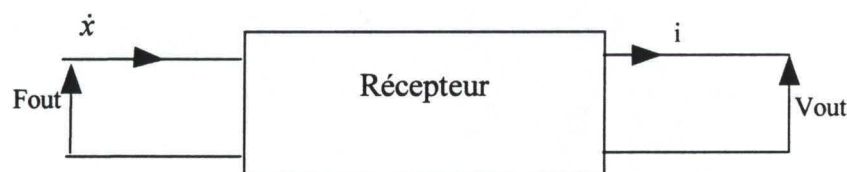


Figure A-3 : Relation directe

La relation entre la vibration due à l'onde reçue et le courant généré par cette dernière est donnée par l'équation A-5:

$$i = d.k.\dot{x} \quad (A-5)$$

On remarque sur le schéma électrique (figure A-4) le couplage entre l'étage d'entrée et de sortie donné par l'équation A-8

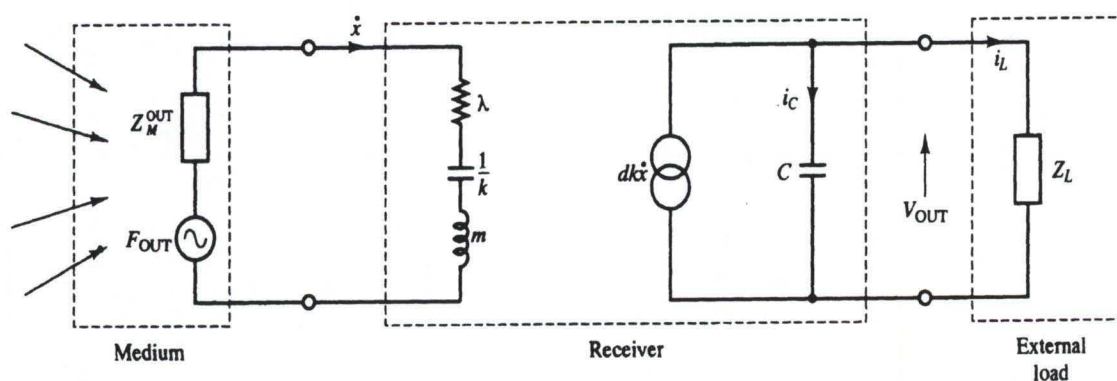


Figure A-4 : Modèle électrique en réception [Bentley95]

Z_{OUT} comme Z_{IN} représente la résistance de l'air et cette dernière est négligeable. Z_L représente l'impédance de la charge branchée aux bornes du transducteur.

$$F_{OUT} = \lambda \cdot \dot{x} + \frac{1}{k \cdot s} \dot{x} + m \cdot s \cdot \dot{x} \quad (A-6)$$

$$V_{OUT} = \frac{Z_L}{1 + Z_L \cdot C} d \cdot k \cdot \dot{x} \quad (A-7)$$

$$\Rightarrow \frac{V_{OUT}}{F_{OUT}} = \frac{d}{C} \frac{\tau \cdot s}{1 + \tau \cdot s} \frac{1}{\frac{1}{w_n^2} s^2 + \frac{2 \cdot \xi}{w_n} s + 1} \quad (A-8)$$

A.1.3. Le modèle électrique d'un transducteur qui détecte un objet

Il s'agit des deux fonctions de transfert trouvées précédemment auxquelles on ajoute les atténuations du signal acoustique dues à la propagation et à la réflexion. Les expressions des atténuations sont données dans le chapitre 2 respectivement en 2-15 et 2-2. Il est possible de négliger le terme dû à la réflexion car ce dernier tend vers 1 lorsque l'objet est dur.

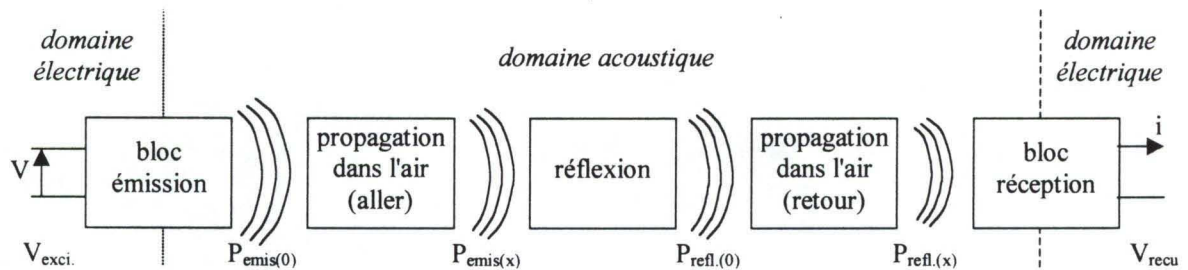


Figure A-5 : Transmission-Reception acoustique

Les deux blocs, synthétisant la propagation dans l'air de l'onde acoustique, correspondent l'atténuation de l'onde pour une distance correspondant à 2 fois la distance entre le transducteur et l'objet détecté. Le gain total est donné par l'équation A-9

$$\frac{V_{recu.}}{V_{exci.}} = \frac{P_{emis(0)}}{V_{exci.}} * \frac{P_{emis(x)}}{P_{emis(0)}} * \frac{P_{refl.(0)}}{P_{emis(x)}} * \frac{P_{refl.(x)}}{P_{refl.(0)}} * \frac{V_{recu.}}{P_{refl.(x)}} \approx \frac{P_{emis(0)}}{V_{exci.}} * 2 * \frac{P_{emis(x)}}{P_{emis(0)}} * \frac{V_{recu.}}{P_{emis(x)}} \quad (A-9)$$

Par comparaison, $V_{\text{exci.}}$ correspond à V dans [Bentley95], V_{recu} correspond à $V_{\text{out.}}$. Dans l'équation précédente, la grandeur acoustique est la pression comparée à la force. La relation entre ces deux grandeurs est l'équation (A-10), avec S la surface de l'élément sensitif.

$$P = \frac{F}{S} \quad (\text{A-10})$$

Certains constructeurs donnent les quelques paramètres électriques de leurs transducteurs. Les paramètres suivants sont ajustés pour correspondre à la forme du signal électrique reçu.

Fréquence d'émission : $f_e = 49,41\text{e}+03 \text{ Hz}$

Pulsation d'émission : $w_n = 2*\pi*49,41\text{e}+03 \text{ rd/s}$

Facteur d'amortissement : $\xi = 0.1$

$d_r = K_r/k_r$: $d_r = 2\text{e-}10 \text{ Coulomb.N}^{-1}$

Coefficient de rigidité du cristal : $k_r = 2\text{e}+09$

Résistance de charge : $R_L = 100\text{e}+03 \Omega$

Capacité du cristal : $C = 1\text{e-}09 \text{ F}$

Constante de temps : $\tau = R_L * C$ en seconde

Pour ajuster notre modèle, il est nécessaire de connaître le coefficient d'absorption. La section suivante donne le calcul du coefficient d'absorption d'après la norme ISO. Elle sera terminée par une application numérique de l'atténuation entre un signal émis et reçu.

A.2. Calcul du coefficient d'absorption de l'air

Extrait de la norme ISO ISO9613-1:1993

But: trouver la variable α_{us} qui est le coefficient d'absorption (en dB/m)

Les variables d'entrée

P : pression ambiante (en Pascal)

T_C : température ambiante (en Celcius)

H : taux d'hygrométrie (en %)

f_e : fréquence du transducteur (en Hz)

Les calculs

$$T_K = T_C + 273,15 \quad A-11$$

$$p = \frac{P}{101325} \quad A-12$$

$$C = 4,6151 - 6,8346 * \left(\frac{273,16}{t} \right)^{1,261} \quad A-13$$

$$h = H * 10^C * p \quad A-14$$

$$tr = \frac{T_K}{293,15} \quad A-15$$

$$frO = p * \left(24 + 4,04 \cdot 10^4 * h * \frac{0,02 + h}{0,391 + h} \right) \quad A-16$$

$$frN = p * \frac{1}{\sqrt{tr}} * \left(9 + 280 * h * e^{-4,17 * \left(\frac{1}{\sqrt{tr}} - 1 \right)} \right) \quad A-17$$

$$\alpha_{us} = 8,686 * f_e^2 * \left(1,84 \cdot 10^{-11} * \frac{1}{p} * \sqrt{tr} + tr^{-2,5} * \frac{0,01275 * e^{\frac{2239,1}{t}}}{frO + \frac{f_e^2}{frO}} + \frac{0,1068 * e^{\frac{33521}{t}}}{frn + \frac{f_e^2}{frN}} \right) \quad A-18$$

A.3. Application numérique de l'atténuation totale (pour plusieurs distances...)

A partir de la norme, il est possible de tracer l'amplitude de l'écho pour un objet se trouvant

à une distance variable (figure A-6). Sur cette figure, l'amplitude est normalisée en fonction de l'amplitude à 1 mètre pour chaque fréquence. On remarque que pour une fréquence plus petite, l'atténuation en fonction de la distance est plus faible que pour une grande fréquence. Cette atténuation est compensée par le gain variable qui est ajusté selon la fréquence.

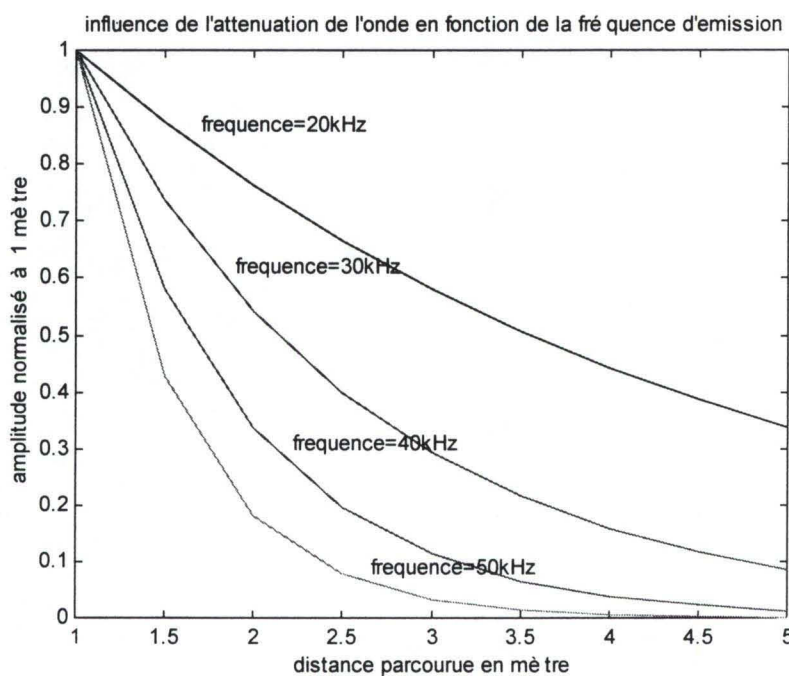


Figure A-6 : Atténuation versus la fréquence

A.4. Comment choisir son transducteur ?

A.4.1. Principe

Une des performances importantes du transducteur est sa sensibilité à l'émission comme à la réception. La sensibilité totale (sensibilité émise sur celle reçue) permet d'évaluer l'amplitude du signal reçu en fonction du type du transducteur et du signal qui le stimule. Pour une même amplitude de sortie, plus le signal a une grande sensibilité, plus le signal de stimulation sera faible.

On peut reprendre cette description en séparant la sensibilité émise de celle reçue. Plus la sensibilité émise est importante, plus l'onde émise portera plus loin pour un même signal de stimulation. Plus la sensibilité reçue est négative faible (en décibel), plus l'onde réfléchie proviendra de loin pour un même niveau de tension de sortie.

A.4.2. Les sensibilités

La sensibilité change en fonction de la distance qui sépare le transducteur de sa cible. Il est important pour pouvoir comparer les différents types de transducteur, d'avoir la même définition de la sensibilité.

La sensibilité à l'émission est définie pour une pression de référence de $20 \cdot 10^{-6}$ Pa à 1 mètre et avec 1 volt efficace d'excitation.

La sensibilité à la réception est définie en volt efficace par pression de référence ($20 \cdot 10^{-6}$ Pa)

Unité rencontrée:

1 bar = 10^5 Pascal ($0,0002 \cdot 10^{-6}$ bar = $20 \cdot 10^{-6}$ Pascal)

10^{-6} bar = $5000 \cdot 20 \cdot 10^{-6}$ Pa, soit un gain de 73.98 dB

On calcule la sensibilité en décibel.

1er exemple: "*transmitting sensitivity = 110 dB re $20 \cdot 10^{-6}$ Pa at 1 meter (300 vac pk-pk, 150 vdc bias)*" [Polatoid 600 series]. Il faut convertir la tension de stimulation par rapport à 1V efficace (rms). La tension efficace de (300vac pk-pk, 150 vdc bias) est 362,13 Volt, il faut enlever $20 \cdot \log_{10}(1/362,13) = 51,18$ dB, La sensibilité pour une excitation de 1Vrms est de 51,18 dB référencée par $20 \cdot 10^{-6}$ Pa à 1 mètre.

2ème exemple: "*transmitting sensitivity = 25 dB vs. 1 μ bar per volt at 1 feet*" [Massa TR89/B type 23]. Il faut convertir la pression 1 μ bar par rapport à $20 \cdot 10^{-6}$ Pa et la distance

en pied en mètre, On ajoute 73,98 dB (conversion μbar) et on soustrait 10,32 dB (conversion des pieds). La sensibilité par 20 μPa pour 1 Volt rms à 1 mètre est de 88,66 dB.

Le "*Sound Pressure Level*" (SPL) donne le gain du signal à 1 mètre du transducteur en tenant compte du signal de stimulation par rapport à une pression de référence.

A.4.3. Les pertes en vol

L'onde en fonction de la fréquence d'émission est atténuée en fonction du trajet. On retrouve le terme TL pour *Transmitting Losses*. La formule de l'atténuation est

$$P(r) = \frac{A(r=1)}{r} e^{-\alpha_{us} \cdot (r-1)} \text{ avec } r \text{ le rayon où se trouve la cible par rapport au transducteur, } \alpha_{us}$$

est le coefficient d'atténuation.

$$TL = 20 \cdot \log_{10} \left(\frac{A(r=1)}{P(r)} \right) = 20 \cdot \log_{10}(r) + 20 \cdot \log_{10}(e^{\alpha_{us} \cdot (r-1)})$$

A.4.4. La sensibilité totale

La sensibilité en réception permet de trouver, à partir de la pression acoustique à 1 mètre du transducteur (excitation + SPL - TL), la tension efficace reçue aux bornes du transducteur.

En multipliant les fonctions de chaque bloc respectif de la figure A-7, on trouve le rapport de sensibilité entre la tension reçue et celle émise pour une distance donnée.

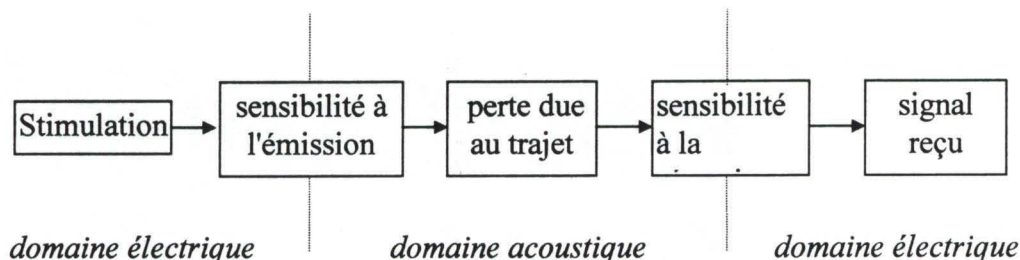


Figure A-7 : Schéma bloc de la sensibilité totale

Application:

On prend un transducteur Massa TR89/B type 40. On stimule le transducteur à l'aide d'une onde sinusoïdale à 40 kHz et d'amplitude 10 Volt crête. L'objet que je souhaite détecter se trouve à 5 mètres.

transmitting sensitivity: 20 dB vs. 1 μ bar per volt at 1 feet

receiving sensitivity: -58 dB vs. 1 Volt per μ bar.

α_{us}/f^2 : $1,37 \cdot 10^{-11}$ Np (air sec)

1) Calcul de la sensibilité à l'émission

1 μ bar = $5000 \cdot 20 \cdot 10^{-6}$ Pa, soit un gain de 73,98 dB

1 feet = 0,3058 mètre, soit un gain de -10,32 dB

La sensibilité à l'émission est 83,66 dB par 20 μ Pa pour 1 Volt rms à 1 mètre

2) Calcul du SPL pour le signal de stimulation

La tension efficace du signal de stimulation est 7,071 V, ce qui correspond à un gain de 16,99 dB. Le SPL est de 100,65 dB par 20 μ Pa à 1 mètre pour le signal d'excitation

3) Calcul du TL pour un objet à 5 mètres

$\alpha_{us} = 1,37 \cdot 10^{-11} \cdot (40 \cdot 10^3)^2 \cdot 8,686 = 0,194$ dB/m

$TL = 20 \cdot \log(5) + 8,7 \cdot \alpha \cdot (5-1) = 7,3536$ dB

4) Bilan des gains et des pertes (réfléchis à 1 mètre)

$$\text{Total} = \text{SPL} - \text{TL} = 100,65 - 7,35 = 93,30 \text{ dB (par rapport à une pression de } 20 \mu\text{Pa)}$$

5) Calcul de la valeur efficace de l'onde reçue

La sensibilité à la réception est en volt par Pascal ($1 \mu\text{bar} = 0,1 \text{ Pa}$)

La pression réfléchie à 1 mètre est $20 \cdot 10^{-6} \text{ Pa} \cdot 10^{(\text{bilan}/20)} = 0,9248 \text{ Pa}$

La valeur efficace du signal reçu est $0,9248 \cdot 10^{(-58/20)} \cdot 10 = 0,0116 \text{ Volt}$

Remarque: Le récepteur se trouve dans l'axe de l'émetteur (orientation = 0 degré)

Annexe B :

Compléments pour l'algorithme de mesure de distance

B.1. Choix des fonctions de prétraitement

Le programme qui teste les différentes fonctions de prétraitement est relativement simple, on effectue le calcul de l'intercorrélation entre la fonction de prétraitement d'entrée et celle de sortie avec un retard programmable (pas de 1/48 période). La position du maximum et du minimum est trouvée à l'aide des commandes min, max et find de Matlab, la moyenne à l'aide de mean et l'écart type à l'aide de std.

```
%*****%
%          programme rep35e.m          %
%          %                             %
%  test des fonctions de prétraitement %
%          %                             %
%  Auteur : Arnaud Lucas de Couville   %
%*****%
%
%  génération d'un sinus échantillonné à 3/4 de T, avec un retard variable
%  puis tester la fonction d'intercorrélation
%
%  emis : sinus de 15 périodes ou fonction porte, échantillonné à 3/4*T
%  reçu : réponse du sinus dans un filtre, cf mod_ult1.m (simulink)

clear
load recept

% la valeur à trouver en échantillon est 20-[0:1] (retard)

emis=ones(1,20);
%emis=sin(0:(3/4*2*pi):15*2*pi-(3/4*2*pi));
recept=[zeros(35,1);recept];
int=[];

for(i=1:36)
    for(j=1:40)          % on acquiesitionne 2n echantillons
        ech(i,j)=recept(-i+1+j*36);
```

```

end
end

for(i=1:36)
    recu=zeros(1,20),ech(i,:);
    ret(i)=i-1;
    phase(i)=360*ret(i)/48;
    in=calinter((emis),abs(recu));

    %résultat de l'intercorrélation pour les échantillons de 0 à n-1

    mt=find(max(in)==in)-1;
    mmt=find(min(in)==in)-1;
    mmmt=find(max(abs(in))==abs(in))-1;
    m(i)=mt(1);
    mm(i)=mmt(1);
    mmm(i)=mmmt(1);
    nm(i)=length(mt);
    nmm(i)=length(mmt);
    nmmm(i)=length(mmmt);

    %interpolation parabolique

    ml=mmm(i)+1;
    y=in(ml-1:ml+1);
    v=y(1);
    u=0.5*(-3*y(1)+4*y(2)-y(3));
    t=y(2)-y(1)-u;

    xmax=-u/(2*t);           % dérivée s'annule
    tmax(i)=xmax+ml-1-1;
end

res=[ret',phase',m',nm',mm',nmm',mmm',nmmm',tmax'];

moy_=mean(mmm);
max_=max(mmm);
min_=min(mmm);
ect_=std(mmm);
tableau=[moy_,max_,min_,ect_]

```

B.2. Influence du nombre de cycles émis

A partir du modèle synthétique (Annexe A), le train d'ondes carrées est filtré dans les fonctions de transfert du transducteur puis est échantillonné à la fréquence d'acquisition. Sur la figure B-1, l'onde ne commence pas à un temps 0, puisque l'on impose un délai de 5 périodes dans le programme.

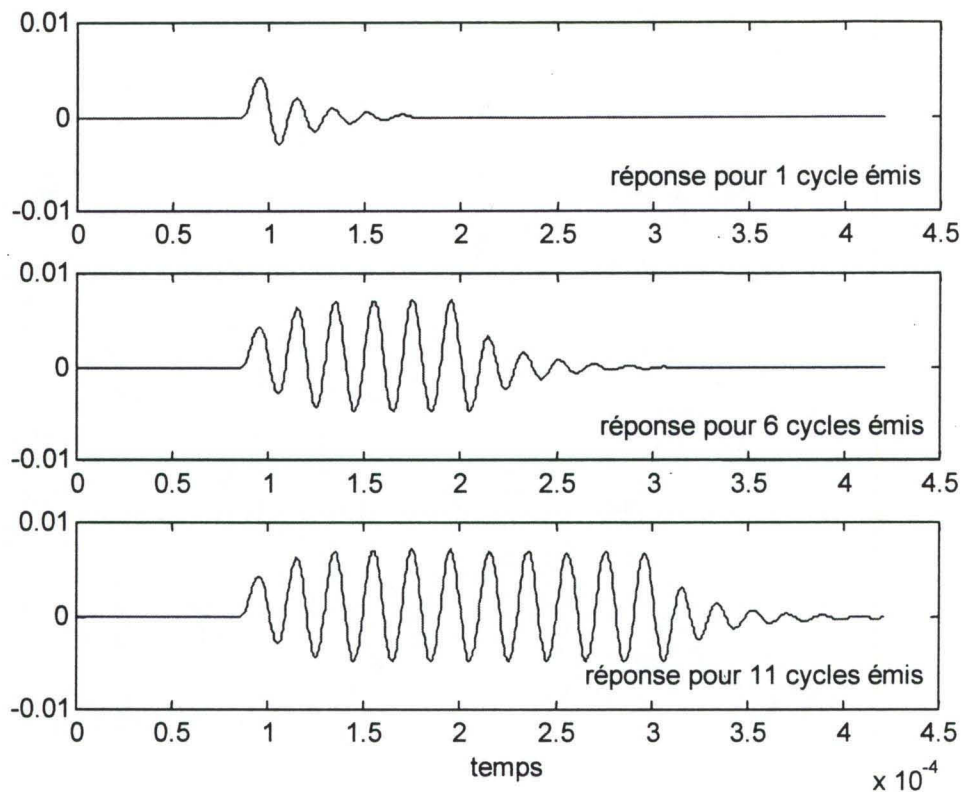


Figure B-1 : Influence du nombre de périodes émises

```

%*****%
%   programme nb_cycle_emis.m   %
%                               %
%   influence du nombre de cycles émis %
%                               %
%   Auteur : Arnaud Lucas de Couville %
%*****%
%
% programme qui génère une onde ultrasonore à partir de nb_emis période
% du signal d'émission
% et test le résultat (intercorrélation) pour un ratio (emis/acq)
% constant
% la fonction et les valeurs des composants proviennent de la doc du
% TL852
%
% fréquence d'acquisition fixe à 4/3 de f_emis

%création du signal

periode=1/49.41e3;
disp('émission de 1 à 20 cycles')

for(retard=1:20)
    nb_per=retard;
    coef_acq=1.33;

```

```

ech=1./(coef_acq.*49.41e3);
phasemax=round(48/coef_acq);
ncoef_acq=phasemax/48;
nech=(ncoef_acq*periode);
[u,t]=gensig('square',periode,nb_per*periode,nech/phasemax);
nb_delaiss=5;
nb_emis=nb_per*periode/nech;
nb_fois=2; %sur combien de longueur de emis on effectue
l'intercorrelation de recu

    tt=[(-nb_delaiss*phasemax:1:-
1)*nech/phasemax,t',(t(length(t))+(1:1:(nb_fois+1)*nb_emis*phasemax)*nech
/phasemax)]+nb_delaiss*nech;
    tt(1)=0; %(evite un warning, erreur de calcul à 1e-20)
    uu=[zeros(1,nb_delaiss*phasemax),u',zeros(1,round((nb_fois+1)*nb_emis*p
hasemax))];

%le transducteur d'après la documentation du TL852 (texas instruments)
R=1.5e+3;
C=2.2e-9;
L=21e-3;
Ct=500e-12;
Rt=490e+3;

num=[R*C 0];
den=[Ct*Rt*C*L (L*C)+(R*Rt*C*Ct) Rt*(C+Ct)+R*C 1];

%la reponse ultrasonore
[ys,ts]=lsim(tf(num,den),uu,tt);

% on fait varier la phase
nb_recu=length(phasemax:phasemax:length(ys));

for(index=1:phasemax)
    emis(index,1:round(nb_emis))=ones(1,round(nb_emis));
    recu(index,1:nb_recu)=ys(index:phasemax:nb_recu*phasemax)';

    % comme le programme du micro-contrôleur
    % avec detection de seuil, on garde nb_fois*n points
    seuil=0.0002;
    n=size(emis,2);
    m=length(recu(index,:));
    for(a=1:m)
        if((recu(index,a)>=seuil)|(recu(index,a)<=-seuil)) % z>=seuil
ou z<=-seuil
            y=abs(recu(index,a-2:a+(nb_fois)*n-1-2));
            pa(index)=a;
            break;
        end
    end

    % Calcul de l'intercorrelation (pour emis = ones(1,n))
    zz=[y zeros(1,n)];
    c=zeros(1,(nb_fois+1)*n-1);
    for k=1:nb_fois*n
        for i=1:n

```

```

        c(k)=c(k)+zz(i+k-1);
    end
end
c=c(1:nb_fois*n);

[val(index),pos(index)]=max(c);
echmax(index)=pos(index)+pa(index)-2-1;
tmax(index)=echmax(index)*ech;

    %sprintf('\n la periode : %0.5g \n nb de periode : %d \n freq. d
acquisition : %5.1f \n delais au debut : %5.3f \n resultat de l
intercorrelation : %d',periode, nb_per, coef_acq/periode, 100e-
6/(periode/coef_acq), echmax(index))
end

    %hold on
    %plot(ts(1:phasemax:nb_recu*phasemax),ys(1:phasemax:nb_recu*phasemax),
'or')

    % les résultats

    mt=find(max(echmax)==echmax);
    mmt=find(min(echmax)==echmax);
    m(retard)=echmax(mt(1));
    mm(retard)=echmax(mmt(1));
    nm(retard)=length(mt);
    nmm(retard)=length(mmt);
    moy(retard)=mean(echmax);

    %sprintf('T=%d \t max \t min \n val: \t %d \t %d \t moyenne: %2.2f \n
nb: \t %d \t %d \n valeur à trouver :
%2.2f',nb_per,m,mm,moy,nm,nmm,nb_delais+1)

end

```

B.3. Influence de la fréquence d'acquisition

Ce programme permet d'évaluer précisément l'impact du choix de la fréquence d'acquisition pour le calcul de l'intercorrélation. Par exemple, si on prend une sinusoïde de phase 0 et que l'on échantillonne à sa période, on aura que des 0 et le résultat sera inexploitable pour déterminer le temps de vol. Il faut trouver un moyen de calculer les différentes moyennes des points acquisitionnés pour différentes phases de départ.

```

%*****%
%      programme freq_acq.m      %
%                               %
%  influence de la fréq. d'acquisition %
%                               %
%  Auteur : Arnaud Lucas de Couville %
%*****%

```

```

%
% programme qui test différentes fréquences d'acquisition
% en vue du calcul de l'intercorrélation
%
% les périodes d'acquisition vont de 1/2 à 1 fois la période
% d'émission par pas de 1/512

t=1/2:1/256:1;
for(index=1:length(t))
    %t=nb_etat_acq/nb_etat_emis;
    nb_acq=16;

    for(i=1/72/3:1/72/3:t(index))    %sinus echantillonné tous les 1/72/3

x(round(i*72*3),1:nb_acq)=sin(i*2*pi:t(index)*2*pi:nb_acq*t(index)*2*pi);
    moy_x=sum(abs(x(round(i*72*3),1:nb_acq))/nb_acq);
    moy(round(i*72*3),:)=moy_x;
    moy_max(index)=max(moy);
    moy_min(index)=min(moy);
    end

    resu(index,1:3)=[t(index),moy_min(index),moy_max(index)];
end

plot(resu(:,1),[resu(:,2),resu(:,3)])

```

B.4. Influence du second écho

Ce programme a pour but de simuler un deuxième écho proche du premier et de déterminer la plage pour laquelle le deuxième écho peut fausser la mesure du temps de vol. Le programme va calculer le temps de vol pour divers cas de second écho (amplitude et retard).

```

%*****%
%      programme deux_echo.m      %
%                                  %
%  influence de la fréq. d'acquisition %
%                                  %
%  Auteur : Arnaud Lucas de Couville %
%*****%
%
%  fichier d'application de l'intercorrélation sur la donnée zo_st_20
%  (mes1.mat) provenant de l'oscilloscope
%  premiere colonne : temps
%  deuxieme colonne : amplitude
%
%  version du 07 juin 98
%  addition d'un nouvelle echo avec délai variable

```



```

%load mes_s1
%load mes_s2
%load mes_s3
%load mes_s4
%load mes_s5
%zo_st_20=[mes_s1;mes_s2;mes_s3;mes_s4;mes_s5];

%save zo_st_20.mat zo_st_20
load zo_st_20.mat

ech=3/4/49.41e3; % periode d'echantillonnage (4/3 de femis)

debut=zo_st_20(1,1);
fin=zo_st_20(length(zo_st_20),1);
pas=zo_st_20(2,1)-zo_st_20(1,1);

x_ech=debut:ech:fin;

deb_ech=ceil(debut/ech);
fin_ech=floor(fin/ech);

y_ech=interp1(zo_st_20(:,1),zo_st_20(:,2),x_ech).*mean(zo_st_20(:,2));

emis=ones(20,1);

nbzero=round(1e-3/ech);
recu=[zeros(1,nbzero),y_ech(nbzero+1:length(y_ech))-
mean(y_ech(nbzero+1:length(y_ech))),zeros(1,40)];

% comme le programme du micro-contrôleur
% avec detection de seuil, on garde 3*n points
seuil=0.0002;
n=length(emis);
m=length(recu);
for(a=1:m)
    if((recu(a)>=seuil)|(recu(a)<=-seuil)) % z>=seuil ou z<=-seuil
        y=abs(recu(a-2:a+3*n-1-2));
        break;
    end
end

% Calcul de l'intercorrelation
zz=[y zeros(size(y))];
c=zeros(1,4*n-1);
for k=1:3*n
    for i=1:n
        c(k)=c(k)+emis(i)*zz(i+k-1);
    end
end
c=c(1:3*n-1);

[val,pos]=max(c);
tmax=pos+a-2-1;

% on ajoute un deuxième echo qui suit d'amplitude multiplié par "mul"
% et décalé de "d" échantillons
% NB: le décalage max. est de 60

```

```

for(mul=1:1:12)
    for(d=1:60)
        multi=mul*0.25;
        drecu=recu+[zeros(1,d),recu(1:length(recu)-d).*multi];
        seuil=0.0002;
        dn=length(emis);
        dm=length(drecu);
        for(da=1:dm)
            if((drecu(da)>=seuil)|(drecu(da)<=-seuil))    % z>=seuil ou z<=-
seuil
                dy=abs(drecu(da-2:da+3*n-1-2));
                break;
            end
        end
        % Calcul de l'intercorrelation
        dzz=[dy zeros(size(dy))];
        dc=zeros(1,4*dn-1);
        for dk=1:3*dn
            for di=1:dn
                dc(dk)=dc(dk)+emis(di)*dzz(di+dk-1);
            end
        end
        dc=dc(1:3*n-1);
        [dval(mul,d),dpos(mul,d)]=max(dc);
        dtmax(mul,d)=dpos(mul,d)+da-2-1;
    end
end

break
%configuration affichage
h=mesh([1:60],[0.25:0.25:3],dtmax);
view(-10,30)
set(h,'EdgeColor',[0,0,0])

```

Annexe C :

Compléments pour l'algorithme d'estimation de la tangente

C.1. Changement de référentiel

But : passer un point du repère mobile $\{M\}$ vers le repère origine $\{O\}$.

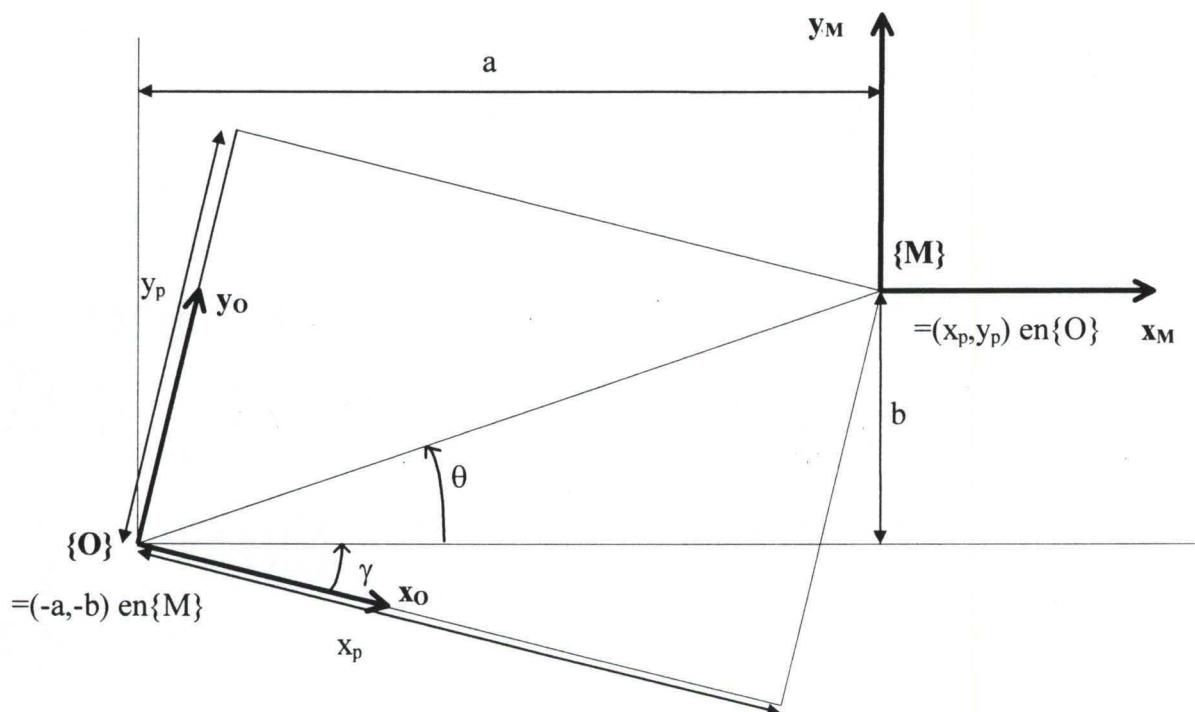


Figure C-1 : Changement de repère

La figure ci-dessus montre le passage d'un repère à un autre. Il est important de connaître la position de l'ancien repère dans le nouveau. Dans le cas de changement de repère $\{M\}$ à $\{O\}$, il faut connaître (x_p, y_p) et l'angle entre les repères, γ . Pour effectuer le changement de

repère, il faut définir l'angle γ entre deux axes (un nouveau par rapport à l'ancien) avec lequel on effectue une rotation d'axe pour avoir les deux axes dans la même direction. Ici, on peut considérer un repère orthogonal en trois dimensions où l'axe Oz pointe vers le lecteur. La matrice de transformation est donnée en C-1.

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & x_p \\ \sin(\gamma) & \cos(\gamma) & 0 & y_p \\ 0 & 0 & 1 & z_p \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} \quad (C-1)$$

Un point M (x_M, y_M) dans le repère $\{M\}$ s'écrira dans le repère $\{O\}$ soit la forme suivante :

$$x = x_M * \cos(\gamma) - y_M * \sin(\gamma) + x_p \quad (C-2)$$

$$y = x_M * \sin(\gamma) + y_M * \cos(\gamma) + y_p \quad (C-3)$$

C.2. Simulateur pour la fusion de données

C.2.1. Création du simulateur

On désire avoir un simulateur qui fonctionne dans toutes les sortes de mur (position et inclinaison). On va utiliser un changement de repère pour simuler les inclinaisons du mur qui correspondra à l'angle γ . La mesure de la position du mur va être identique quelque soit l'inclinaison, puisqu'on prend le cas d'un mur vertical dans le repère du système mobile $\{M\}$.

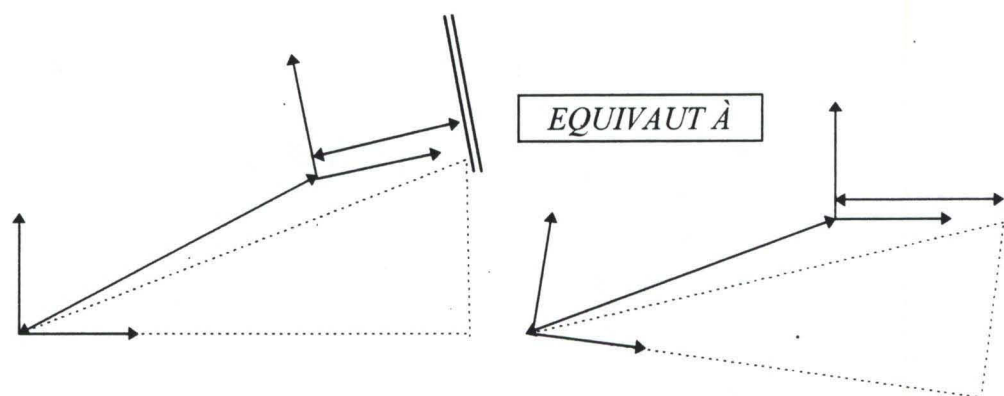


Figure C-2 : Transformation géométrique pour le simulateur

En utilisant les notations de la figure C-1, on peut déterminer le vecteur de changement de position du référentiel $\{M\}$ dans le référentiel $\{O\}$, $[x_p, y_p, 0]^T$.

$$d = \sqrt{a^2 + b^2} \quad (C-4)$$

$$\theta = \arctan(b/a) \quad (C-5)$$

$$x_p = d \cdot \cos(\theta + \gamma) \quad (C-6)$$

$$y_p = d \cdot \sin(\theta + \gamma) \quad (C-7)$$

C.2.2. Générateur de trajectoire

Programme: genetraj.m

But: Pour toujours simplifier notre simulateur, nous allons utiliser une sinusoïde sur une demi-période (équivalent de 0 à π) et nous allons exploiter les propriétés mathématiques de cette fonction.

On peut calculer une trajectoire sur $[0, T/8]$ et trouver la trajectoire sur $[T/8, T/4]$ par simple symétrie centrale en $T/8$. De même façon, on trouve la portion de la trajectoire $[T/4, T/2]$ par une symétrie d'axe ($x=T/4$) avec la trajectoire sur $[0, T/4]$.

x est un vecteur commençant par 0 jusqu'à 5.5 par pas de 0.5

y est un vecteur correspondant à $a(i)+y(i)$, comme le vecteur commence à $i=0$, on ajoute la dernière valeur y trouvée

ex: $i=1$ $x = [0 \ 0.5]$
 $y = [0 \ a(1)+y(1)] = [0 \ 0+0] = [0 \ 0]$
 $i=2$ $x = [0 \ 0.5 \ 1]$
 $y = [0 \ 0 \ a(2)+y(2)] = [0 \ 0 \ 0+0] = [0 \ 0 \ 0]$
 $i=3$ $x = [0 \ 0.5 \ 1 \ 1.5]$
 $y = [0 \ 0 \ 0 \ a(3)+y(3)] = [0 \ 0 \ 0 \ 0.025+0] = [0 \ 0 \ 0 \ 0.025]$
 $i=4$ $x = [0 \ 0.5 \ 1 \ 1.5 \ 2]$
 $y = [0 \ 0 \ 0 \ 0.025 \ a(4)+y(4)] = [0 \ 0 \ 0 \ 0.025 \ 0.025+0.025] = [0 \ 0 \ 0 \ 0.025 \ 0.05]$
 (...)

A l'aide de permutation, on peut déduire x allant de 6 à 11.5 à partir de x(0 à 0.5) (figure C-3 -a-)

Puis avec une autre permutation, on peut déduire x allant de 12 à 23.5 à partir de x(0 à 11.5) (figure C-3 -b-)

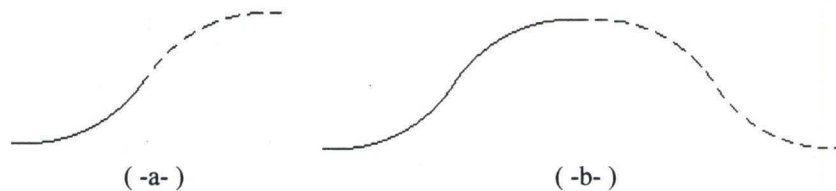


Figure C-3 : Déduction de la trajectoire par permutations

On garde tous les points pairs $(0.5+k)$, car leur pente est connue, $a(\text{précédant})$ et $a(\text{courant})$ sont identiques, les points gardés pour la simulation sont notés sous forme du couple (x_0, y_0)

On calcule l'angle de la direction du robot par rapport à l'axe Ox.

$$\text{pente}(i) = \Delta y / \Delta x = y(i+1) - y(i-1), \quad (\text{C-8})$$

$$\text{angle} = \tan^{-1}(\text{pente}) \quad (\text{C-9})$$

C.2.3. Générateur des positions des deux transducteurs

Programme : genepostr.m

But: En gardant la disposition des transducteurs de l'article de [Song96], il est possible de tracer des deux transducteurs, quelque soit la pente, la distance entre l'axe central et le transducteur est constante ($=d$). dx et dy varie en fonction de la pente.

$$dy = \cos(\text{pente}) * d \quad (C-10)$$

$$dx = \sqrt{d^2 - dy^2} \quad (C-11)$$

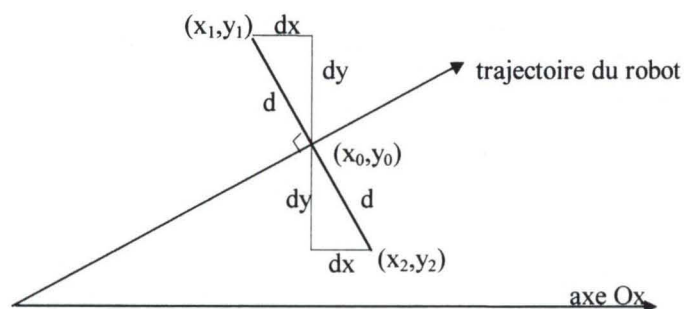


Figure C-4 : Positionnement des transducteurs

$$x_1 = x_0 (+/-) dx \quad (C-12)$$

$$y_1 = y_0 + dy \quad (C-13)$$

$$x_2 = x_0 (-/+) dx \quad (C-14)$$

$$y_2 = y_0 - dy \quad (C-15)$$

Le test de l'angle permet de savoir quel transducteur est en avance sur l'autre (celui qui est en avance, on additionne dx à x_0 ; celui qui est en retard, on soustrait dx à x_0)

C.2.4. Générateur de données ultrasonores

Programme: genedonus.m

But: Compte tenu que l'on se trouve dans le cas (a) de l'article, une perpendiculaire au mur passe par le centre du transducteur et l'origine de cette perpendiculaire se trouve sur l'étendue de mesure du transducteur. Le point mesuré sur le mur aux coordonnées suivantes (x_{mur}, y_i) correspond à l'intersection entre la distance mesurée la plus courte et le mur.

Maintenant, on cherche à déterminer le point d'impact imaginaire sur l'axe du transducteur (tangente à la trajectoire du système) pour la distance mesurée d , les coordonnées sont (x_{di}, y_{di}) . Pour le point sur le mur sur cet axe, il faudra trouver la nouvelle distance ainsi que les coordonnées (x_{mur}, y_{bi}) .

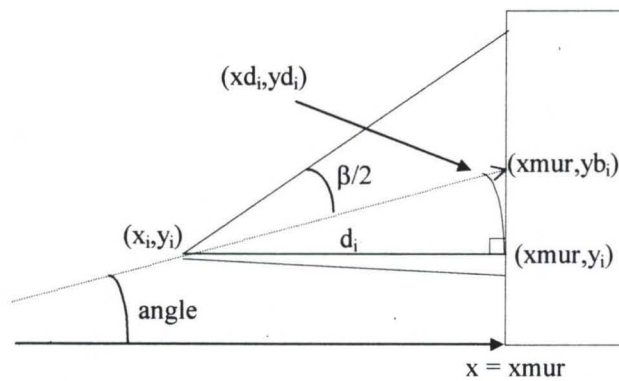


Figure C-5 : Mesure de distance transducteur-mur

La droite passant par le centre du transducteur et par sa normale aura la même pente que la trajectoire du robot, il suffit de trouver le coefficient b_i de la droite ($y_i = a.x_i + b_i$)

En considérant un triangle rectangle de sommets (x_{di}, y_{di}) et (x_i, y_i) , on peut appliquer la trigonométrie pour trouver la valeur x_{di} (C-17). Pour trouver le point d'impact fictif entre l'axe du transducteur et le mur, on trouvera y_{bi} en remplaçant x de l'équation de la droite par x_{mur} , la distance dd_i sera trouvée à l'aide du théorème de Pythagore (C-18)

$$x_{di} = x_i + d_i \cdot \cos(\text{angle}(i)) \quad (\text{C-17})$$

$$dd_i^2 = (x_{mur} - x_i)^2 + (y_{bi} - y_i)^2 \quad (\text{C-18})$$

C.2.5. Initialisation du simulateur

Programme : inisimu.m

Au début de chaque lancement du simulateur, l'utilisateur est invité à définir le mouvement du système sensoriel mobile. L'utilisateur a trois possibilités:

Le système détecte un seul mur et l'utilisateur peut choisir le nombre de trajets (le système se déplace vers le mur et s'éloigne du mur) au demi-trajet près.

Le système détecte de façon séquentielle deux murs. L'utilisateur choisit un nombre entier de trajets pour le premier mur et a le choix pour le dernier mur de finir au demi-trajet près.

Le système détecte successivement trois murs. Le système doit terminer son parcours pour les deux premiers murs par l'origine (nombre de trajets entier) et pour le dernier, l'utilisateur a le choix d'arrêter le système à l'origine ou contre le mur (au demi-trajet près).

Pour chaque mur, l'utilisateur donne l'angle gamma d'inclinaison du mur, pour un mur vertical $\gamma = 0$ ou π (mur se trouve dans la direction opposée de l'axe [Ox] du référentielle origine).

Compte tenu du choix de l'utilisateur, le vecteur de position du repère mobile {M} dans le référentiel {O} devra être recalculé à chaque nouveau mur. Les coordonnées x_p , y_p sont calculées uniquement pour un aller vers le mur; pour le retour, les coordonnées seront déduites par simple permutation.

C.2.6. Vérification du choix des facteurs d'oubli

Programme : factverif.m

But: Autoriser la simulation en cas de respect des conditions sur les facteurs pour obtenir des bonnes estimations des paramètres du filtre [Salgado88]

Les conditions sur les facteurs sont expliquées dans le paragraphe sur l'algorithme de Kalman avec facteur d'oubli et de mise à jour dans le chapitre 3

Le programme vérifiera les conditions. En cas de non-respect, il fournira une erreur avec la condition non respectée.

C.2.7. Estimation des paramètres

Lorsque le simulateur a terminé, il donne les valeurs estimées pour chaque donnée de mesure (p_i , q_i). En fonction de l'inclinaison du mur, on peut calculer de façon théorique les deux paramètres, avant de passer au calcul. La figure ci-dessous explique la situation:

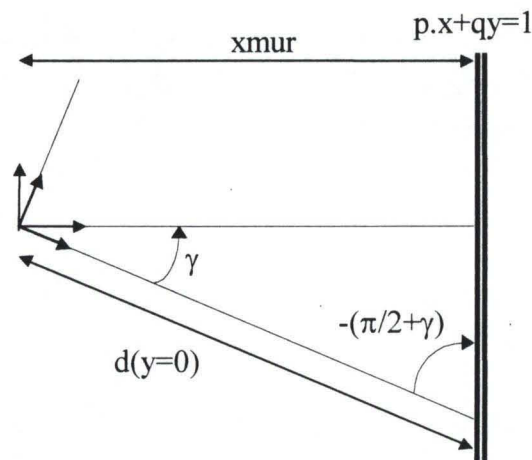


Figure C-6 : Paramètres et leur environnement

En faisant une analogie avec l'équation de droite "classique": $y = a.x + b$, avec a la pente de la droite qui correspond à la tangente de l'angle formé par la droite et l'axe des abscisses au point $x(y=0)$.

Ici nous estimons une équation de droite plus rigoureuse, $p.x + q.y = 1$ et on peut facilement identifier les paramètres a et b (C-19 à C-21)

$$p.x + q.y = 1$$

(C-19)

$$\frac{p}{q}.x + y = \frac{1}{q} \quad (C-20)$$

$$y = -\frac{p}{q}.x + \frac{1}{q} \quad (C-21)$$

Dans le contexte de ce travail, si $y=0$ alors $x=1/p$, la mesure x correspond sur le graphique à une droite d'équation $y=0$. En appliquant la trigonométrie, on trouve p en (C-22)

$$p = \frac{\cos(\gamma)}{x_{mur}} \quad (C-22)$$

On peut trouver la pente, le coefficient a dans la droite "classique":

En identifiant avec le terme équivalent dans la nouvelle notation, on a $a = -p/q$, on trouve le coefficient q (C-23) en utilisant les égalités trigonométriques suivantes (C-24)

$$q = p \cdot \tan(\gamma) = \frac{\cos(\gamma) \cdot \tan(\gamma)}{x_{mur}} \quad (C-23)$$

$$\tan(-(\pi/2 - \gamma)) = -\tan(\pi/2 - \gamma) = -1/\tan(\gamma) \quad (C-24)$$

C.2.8. Choix des algorithmes

L'utilisateur a en sa disposition 7 types d'algorithmes qui sont présentés dans le chapitre 3.

Les choix sont les suivants :

Le filtre de Kalman de l'article de Song pour le système linéarisé.

⇒ programme : kalman_song.m

Le filtre de Kalman de l'article de Song adapté avec les facteurs d'oubli exponentiel et réinitialisation de la matrice de covariance de l'erreur pour le système linéarisé.

⇒ programme : kalman_song_salgado.m

L'algorithme RLS simple pour un système linéaire.

⇒ programme : rls_simple.m

L'algorithme RLS avec un facteur d'oubli pour un système linéaire.

⇒ programme : rls_lambda.m

L'algorithme RLS avec une réinitialisation périodique de la matrice de covariance de l'erreur pour un système linéaire.

⇒ programme : rls_reset.m

L'algorithme RLS avec les facteurs d'oubli exponentiel et réinitialisation de la matrice de covariance de l'erreur pour un système linéaire.

⇒ programme : rls_salgado.m

L'algorithme RLS IMCEVI pour un système linéaire et ajusté pour l'application.

⇒ programme : rls_IMCEVI.m

C.3. Les programmes Matlab

C.3.1. Programme principal : *simulateur.m*

```
% *****%
%
%      Algorithme de fusion      %
%      (chapitre 3.2)           %
%
%      Programmme principal      %
%
% Auteur: Arnaud Lucas de Couville %
% Date: 04 octobre 1998           %
% Version: finale                 %
% *****%
%
% appel : simulateur
%
% préalablement faire un "hold on" dans les figure(2) et figure(3)

clear all
clear function

multi=0.2    %espacement entre 2 points (en mètre)
%multi=1

[x0,y0,theta,pente]=genetraj(multi);

%zone des constantes
taille=length(x0);
d=0.15;      % espacement entre un transducteur et le milieu des 2
transducteurs
sd=2*d;
```



```

xmur=taille*multi; % 24 points tous les 20 cm.

[x1,x2,y1,y2]=genepostr(x0,y0,d,theta);

[d1,d2,dmur1,dmur2,b1,b2,yb1,yb2]=genedonus(xmur,theta,pente,x1,x2,y1,y2,
x0,y0,d);
hold off, close,

pos_tr=[x1;y1;x2;y2];
dmur=[d1;d2];

[xp,yp,b,gammarep,trajet,reponse]=inisimu(dmur,x0,y0,pos_tr,theta);
save inisemi
%load inisemi

%variance=2e-5;
variance=2e-5*(0.2);
bruit=1;

b0=b;
[b,varb]=inikal(b,bruit,variance);
%load inikal

disp(' ')
disp('***** Choix du type d algorithme *****')
disp('1) Filtre de Kalman version Song96');
disp('2) Filtre de Kalman version Song96 et Salgado88 adaptée');
disp('3) RLS (système linéaire)');
disp('4) RLS avec un facteur d oubli (système linéaire)');
disp('5) RLS avec reinitialisation périodique de P (système linéaire)');
disp('6) RLS de Salgado88 (système linéaire)');
disp('7) RLS dédié pour 1 application (système linéaire)');
choix=input('-> -> -> Entrer votre choix : ');
disp('')

switch(choix)
case(1),
    nb_ite=1;
    P0=20*eye(2);

[ae]=kalman_song(nb_ite,b,varb,sd,xp,yp,reponse,trajet,gammarep,taille,P0
);
case(2),
    nb_ite=15;
    P0=20*eye(2);
    alpha=0.3;
    betha=0.5;
    lambda=0.85;
    delta=5e-4;
    [foubli]=factverif(alpha,lambda,betha,delta,P0);

[ae]=kalman_song_salgado(nb_ite,b,varb,sd,xp,yp,reponse,trajet,taille,gam
marep,foubli,P0);
case(3),
    nb_ite=1;
    P0=20*eye(2);
    [ae]=rls_simple(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille);

```

```

case(4),
    nb_ite=4;
    P0=20*eye(2);
    lambda=0.95;

[ae]=rls_lambda(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,lambda);
case(5),
    nb_ite=5;
    P0=20*eye(2);
    treset=5;

[ae]=rls_reset(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,treset);
case(6),
    nb_ite=10;
    P0=20*eye(2);
    alpha=0.6;
    betha=0.1;
    lambda=0.80;
    delta=3e-5;
    [foubli]=factverif(alpha,lambda,betha,delta,P0);

[ae]=rls_salgado(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,foubli);
case(7),
    nb_ite=1;
    P0=5*eye(2);
    lambda=0.96;
    immu=15;
    seuil=0.10;
    biais=15e-3;

[ae]=rls_imcevi(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,immu,lambda,seuil,biais);
otherwise,
    disp('mauvais choix');
    break
end

size(ae)

figure(3)
plot(ae(2,:), 'c')

figure(2)
plot(ae(1,:), 'c')

%break

%calcul des paramètres théoriques
theo=[];
gammatot=[];
reptheo1=ones(1,trajet(1)*taille)*cos(gammarep(1))/xmur;
reptheo2=reptheo1*tan(gammarep(1));
gammatot1=ones(1,trajet(1)*taille).*gammarep(1);
theo=[0 reptheo1;0 reptheo2];

```

```

    gammatot=[0 gammatot1];
    if((reponse==2)|(reponse==3))
        reptheol=ones(1,trajet(2)*taille)*cos(gammarep(2))/xmur;
        reptheo2=reptheol*tan(gammarep(2));
        gammatot2=ones(1,trajet(2)*taille).*gammarep(2);
        theo=[theo [reptheol;reptheo2]];
        gammatot=[gammatot gammatot2];
        if(reponse==3)
            reptheol=ones(1,trajet(3)*taille)*cos(gammarep(3))/xmur;
            reptheo2=reptheol*tan(gammarep(3));
            gammatot3=ones(1,trajet(3)*taille).*gammarep(3);
            theo=[theo [reptheol;reptheo2]];
            gammatot=[gammatot gammatot3];
        end
    end

figure(2)
plot(theo(1,:), 'k-.' )

figure(3)
plot(theo(2,:), 'k-.' )

err(1,:)=abs(ae(1,:)-theo(1,:));
err(2,:)=abs(ae(2,:)-theo(2,:));
[mean(err(1,:)),mean(err(2,:))]

%break

% calcul des parametres de façon directe (modèle géométrique avec bruit)
%rappel:
% (xati,yati) coordonnée entre le centre des 2 transducteurs et le
% transducteur i dans le repère fixe
% (xdmuri,ydmuri) coordonnée de la distance détectée par le
% transducteur i dans le repère fixe
% (xp,yp) coordonnée du centre des 2 transducteurs dans le repère fixe
% (xmurbi,ymurbi) coordonnée du mur détecté par le transducteur i dans
% le repère fixe

[xat1,yat1,xat2,yat2,xp,yp,xdmur1,ydmur1,xdmur2,ydmur2]=modgeo(x0,y0,x1,y
1,x2,y2,gammarep,trajet,b,reponse);

xmurb1=xdmur1+xat1+xp;
ymurb1=ydmur1+yat1+yp;
xmurb2=xdmur2+xat2+xp;
ymurb2=ydmur2+yat2+yp;

%calcul des coefficients de la tangente par le modèle mathématique bruité
for(bi=2:length(xmurb1))
    ae_geo(:,bi)=inv([xmurb1(bi) ymurb1(bi);xmurb2(bi) ymurb2(bi)])*[1;1];
end
err_geo(1,:)=abs(ae_geo(1,:)-theo(1,:));
err_geo(2,:)=abs(ae_geo(2,:)-theo(2,:));
[mean(err_geo(1,:)),mean(err_geo(2,:))]

figure(2)
plot(ae_geo(1,:), 'b--')

```

```
figure(3)
plot(ae_geo(2,:), 'b--')
```

C.3.2. Sous-programme : *genetraj.m*

```
%programme genetraj.m
%
% appel: fonction [x0,y0,theta,pente]=genetraj(multi)
%
% but: génération de point de trajectoire (x0,y0) d'un système allant
%      vers un mur situé à 24 unités et dont la pente de la trajectoire
%      du robot ne dépasse pas 15 degrés
%
% entrée :
% - multi : coef de conversion, donne l'espacement entre 2 mesures (en
% mètre)
% sorties :
% - x0,y0 : coordonnées en (x,y) du milieu des 2 transducteurs
% - theta : angle entre le repère entre le centre des transducteur et
le repère mobile
% - pente : pente à chaque point du milieu des 2 transducteurs
% exprimée par rapport au repère mobile
%
% date: octobre 1998, version final
%
function [x0,y0,theta,pente]=genetraj(multi);

% generation d'une trajectoire bidon, de façon à être légèrement
% décalé et avec un angle inférieure à 15 degrés

x=0;
y=0;
a=[0 0 0.025 0.025 0.05 0.05 0.075 0.075 0.1 0.1 0.125 0.125];
for(i=1:length(a))
x=[x i/2];
y=[y a(i)+y(i)];
end

x=[x [length(x)/2:0.5:length(a)]];
y=[y -fliplr(y(1:length(y)-1))+2*y(length(y))];

x=[x length(x)/2+x(1:length(x)-1)];
y=[y fliplr(y(1:length(y)-1))];

figure(1)
plot (x,y)

hold on

% tracer des points de simulation

x0=x(2:2:length(x))*multi;
y0=y(2:2:length(y))*multi;
```



```

plot(x0,y0,'o')

% calcul de l'angle de chaque points de simulation

pente=[];
for(i=2:2:length(y))
pente=[pente y(i+1)-y(i-1)];
end

theta=atan2(pente,ones(1,length(pente)));

```

C.3.3. Sous-programme : *genepostr.m*

```

% programme genepostr.m
%
% appel: [x1,x2,y1,y2]=genepostr(x0,y0,d,theta)
%
% But: GENERer les POSitions des deux TRansducteurs dans la configuration
%      de l'article de [Song96]
% entrées:
% - x0,y0 : coordonnées en (x,y) du milieu des 2 transducteurs
% - theta : angle entre le repère entre le centre des transducteur et
% le repère mobile
% - d : distance entre le repère mobile et un transducteur
% sorties
% - x1,y1 : coordonnées en (x,y) du transducteur 1
% - x2,y2 : coordonnées en (x,y) du transducteur 2
%
function [x1,x2,y1,y2]=genepostr(x0,y0,d,theta)

% tracer des deux transducteurs

dy=d.*cos(theta);
dx=sqrt(d^2-dy.^2);

for(i=1:length(theta))
    y1(i)=y0(i)+dy(i);
    y2(i)=y0(i)-dy(i);
    if(theta(i)>0)
        x1(i)=x0(i)-dx(i);
        x2(i)=x0(i)+dx(i);
    else
        x1(i)=x0(i)+dx(i);
        x2(i)=x0(i)-dx(i);
    end
end

plot(x1,y1,'bo',x2,y2,'ro')

```

C.3.4. Sous-programme : *genedonus.m*

```

% programme genedonus.m

```

```

%
% appel: [d1,d2,dmur1,dmur2,b1,b2,yb1,yb2]=genedonus(theta,pente,x1,x2,
% y1,y2,x0,y0,d)
%
% But: GENERer les DONnées provenant des deux transducteurs UltraSonores,
%       c'est à dire la distance mesuré par le transducteur
%
% entrées:
% - x0,y0 : coordonnées en (x,y) du milieu des 2 transducteurs
% - x1,y1 : coordonnées en (x,y) du transducteur 1
% - x2,y2 : coordonnées en (x,y) du transducteur 2
% - theta : angle entre le repère entre le centre des transducteur et
% le repère mobile
% - pente : pente de la normale aux transducteurs
% sorties:
% - b1,b2 : coefficient b de la droite y=ax+b pour le transducteur i
% - d1,d2 : distance la plus courte du mur mesurée par le transducteur
% i
% - dmur1,dmur2 : distance du mur sur la normale du transducteur i
% - yb1,yb2 : ordonnée de l'intersection entre le mur et la normale du
% transducteur i
%
function [d1,d2,dmur1,dmur2,b1,b2,yb1,yb2]=genedonus(xmur,theta,pente,x1,
x2,y1,y2,x0,y0,d)

%ymur0=y0;
%ymur1=y1;
%ymur2=y2;

xb=xmur*ones(size(x0));
for(i=1:length(x0))
    b0(i)=y0(i)-pente(i)*x0(i);
    b1(i)=y1(i)-pente(i)*x1(i);
    b2(i)=y2(i)-pente(i)*x2(i);
    yb0(i)=pente(i)*xmur+b0(i);
    yb1(i)=pente(i)*xmur+b1(i);
    yb2(i)=pente(i)*xmur+b2(i);

    d1(i)=-x1(i)+xmur;
    dmur1(i)=sqrt((xmur-x1(i))^2+(yb1(i)-y1(i))^2);
    xd1(i)=x1(i)+d1(i)*cos(theta(i));
    yd1(i)=pente(i)*xd1(i)+b1(i);

    d2(i)=-x2(i)+xmur;
    dmur2(i)=sqrt((xmur-x2(i))^2+(yb2(i)-y2(i))^2);
    xd2(i)=x2(i)+d2(i)*cos(theta(i));
    yd2(i)=pente(i)*xd2(i)+b2(i);

    d0(i)=-x0(i)+xmur;
    dmur0(i)=sqrt((xmur-x0(i))^2+(yb0(i)-y0(i))^2);
    xd0(i)=x0(i)+d0(i)*cos(theta(i));
    yd0(i)=pente(i)*xd0(i)+b0(i);

    plot([x1(i),xd1(i)], [y1(i),yd1(i)], 'b--')
    plot([x2(i),xd2(i)], [y2(i),yd2(i)], 'r--')
end

```

```

plot(xd1,yd1,'bx')
plot(xb,yb1,'b+')
plot(xd2,yd2,'rx')
plot(xb,yb2,'r+')

plot(xd0,yd0,'k')

```

C.3.5. Sous-programme : inisimu.m

```

% programme inisimu.m
%
% appel: [xp,yp,b,gamma,trajet,reponse]=inisimu(d1,d2,x0,y0,theta)
%
% But: Effectuer l'initialisation de l'environnement pour l'estimation
% des parametres du mur.
%
% entrées:
% - x0,y0 : coordonnées en (x,y) du milieu des 2 transducteurs
% - d1,d2 : distance la plus courte du mur mesurée par le transducteur
% i
% - theta : angle entre le repère entre le centre des transducteur et
% le repère mobile
% sorties:
% - b : vecteur de mesure
% - gamma : inclinaison du mur
% - trajet : nombre d'aller et de retour pour chaque inclinaison
% - reponse: nombre de mur (inclinaison)
%
function [xp,yp,b,gamma,trajet,reponse]=inisimu(dmur,x0,y0,pos_tr,
theta);

x1=pos_tr(1,:);
y1=pos_tr(2,:);
x2=pos_tr(3,:);
y2=pos_tr(4,:);
d1=dmur(1,:);
d2=dmur(2,:);

save inisimul808
load inisimul808

b=[d1;d2;theta];

taille=length(d1);
% questionne l'utilisateur
xxx=1;
while(xxx==1)
    rep=input(' un mur, taper 1\n 2 murs, tapez 2\n 3 murs, taper 3\n
votre choix: ');
    switch(rep)

        case(1)
            reponse=1;
            trajet(1)=input('\ntapez le nombre de trajet vers le mur1 (1/2

```

```

trajet accepté): ');
    gamma(1)=input('entrer 1 inclinaison du mur (en rad): ');
    b1=[];
    trajet(1)=round(trajet(1)*2);
    for(tral=1:trajet(1))
        if((-1)^tral==1)
            b1=[b1 fliplr(b)];
        else
            b1=[b1 b];
        end
    end
    disp('choix: 1 mur')
    xxx=0;
    b=b1;
case(2)
    reponse=2;
    disp('choix: 2 murs')
    trajet(1)=input('\ntapez le nombre de trajet (aller retour) vers le
mur1: ');
    gamma(1)=input('entrer 1 inclinaison du mur (en rad): ');
    b1=[];
    for(tral=1:ceil(trajet(1)))
        b1=[b1 [b, fliplr(b)]]; % 2 murs
    end
    trajet(1)=trajet(1)*2;

    trajet(2)=input('\ntapez le nombre de trajet vers le mur2 (1/2
trajet accepté): ');
    gamma(2)=input('entrer 1 inclinaison du mur (en rad): ');
    b2=[];
    trajet(2)=round(trajet(2)*2);
    for(tra2=1:trajet(2))
        if((-1)^tra2==1)
            b2=[b2 fliplr(b)];
        else
            b2=[b2 b];
        end
    end
    b=[b1,b2];

    xxx=0;

case(3)
    reponse=3;
    trajet(1)=input('\ntapez le nombre de trajet (aller retour) vers le
mur1: ');
    gamma(1)=input('entrer 1 inclinaison du mur (en rad): ');
    b1=[];
    for(tral=1:ceil(trajet(1)))
        b1=[b1 [b, fliplr(b)]]; % 2 murs
    end
    trajet(1)=trajet(1)*2;

    trajet(2)=input('\ntapez le nombre de trajet (aller retour) vers le
mur2: ');
    gamma(2)=input('entrer 1 inclinaison du mur (en rad): ');
    b2=[];

```



```

    for(tra2=1:ceil(trajet(2)))
        b2=[b2 [b,fliplr(b)]]; % 2 murs
    end
    trajet(2)=trajet(2)*2;

    b3=[];
    trajet(3)=input('\ntapez le nombre de trajet vers le mur3 (1/2
trajet accepté): ');
    gamma(3)=input('entrer l inclinaison du mur (en rad): ');
    trajet(3)=round(trajet(3)*2);
    for(tra3=1:trajet(3))
        if((-1)^tra3==1)
            b3=[b3 fliplr(b)];
        else
            b3=[b3 b];
        end
    end
    b=[b1,b2,b3];
    disp('choix: 3 murs')
    xxx=0;

    otherwise
        disp('mauvais choix, recommencer:')
        disp(' ')
        xxx=1;

    end
end

% changement de repere
% (xp,yp) coordonnée du centre des 2 transducteurs dans le repère fixe

xp1=[];xp2=[];xp3=[];
yp1=[];yp2=[];yp3=[];

omega=atan2(y0,x0);
domega=sqrt(x0.^2+y0.^2);

for(ii=1:trajet(1))
    xp10=domega.*cos(omega+gamma(1).*ones(1,length(x0)));
    yp10=domega.*sin(omega+gamma(1).*ones(1,length(x0)));
    if((-1)^ii==(-1))
        %impair%
        xp1=[xp1 xp10];
        yp1=[yp1 yp10];
    else
        %paire%
        xp1=[xp1 fliplr(xp10)];
        yp1=[yp1 fliplr(yp10)];
    end
end

if((reponse==2)|(reponse==3))
    xp20=domega.*cos(omega+gamma(2).*ones(1,length(x0)));
    yp20=domega.*sin(omega+gamma(2).*ones(1,length(x0)));
    for(ii=trajet(1)+1:trajet(2)+trajet(1))

```

```

        if((-1)^ii==(-1))
            %impaire%
            xp2=[xp2 xp20];
            yp2=[yp2 yp20];
        else
            %paire%
            xp2=[xp2 fliplr(xp20)];
            yp2=[yp2 fliplr(yp20)];
        end
    end
end

if(reponse==3)
    xp30=domega.*cos(omega+gamma(3).*ones(1,length(x0)));
    yp30=domega.*sin(omega+gamma(3).*ones(1,length(x0)));
    for(ii=trajet(1)+trajet(2)+1:trajet(3)+trajet(2)+trajet(1))
        if((-1)^ii==(-1))
            %impaire%
            xp3=[xp3 xp30];
            yp3=[yp3 yp30];
        else
            %paire%
            xp3=[xp3 fliplr(xp30)];
            yp3=[yp3 fliplr(yp30)];
        end
    end
end
end
end

xp=[0 xp1 xp2 xp3];
yp=[0 yp1 yp2 yp3];

b=[[0;0;0],b];

```

C.3.6. Sous-programme : inikal.m

```

% programme inikal.m
%
% appel: [b,varb]=inikal(b,bruit,variance)
%
% But: initialise les paramètres de simulation du filtre de Kalman
%
% entrées:
% - b      : vecteur de mesure
% - variance : variance du bruit sur la mesure de la distance
% - bruit   : multiplicateur du bruit
% sorties:
% - b      : vecteur de mesure bruitée
% - varb    : variance de chaque élément du vecteur b
%
function [b,varb]=inikal(b,bruit,variance);

%bruit est un multiplicateur du bruit blanc de variance voulue
% si bruit = 0, les signaux de mesure ne sont pas bruités.
% si bruit = 2, les signaux de mesure sont bruités par un bruit blanc

```

```

d'eccart type double

vard1=variance;
vard2=variance;
varangle=variance*0.1*pi/360/1000; % mesure tres peu bruitée, moteur pas
à pas

varb=[vard1;vard2;varangle];

long_b=length(b);
randn('seed',sum(100*clock)) %truc pour avoir un nombre plus aléatoire

alea1=randn(1,long_b);
alea2=randn(1,long_b);
alea3=randn(1,long_b);

alea1=alea1-mean(alea1);
alea2=alea2-mean(alea2);
alea3=alea3-mean(alea3);

alea1=alea1/sqrt(cov(alea1));
alea2=alea2/sqrt(cov(alea2));
alea3=alea3/sqrt(cov(alea3));

bruit_b=[alea1*vard1*bruit;alea2*vard2*bruit;alea3*varangle*bruit];

b=b + bruit_b;

%save inikal b varb

```

C.3.7. Sous-programme : factverif.m

```

% programme factverif.m
%
% factverif(alpha,lambda,betha,delta,P0)
%
% But : verifier les conditions sur le choix des facteurs d'oubli
% et sur la matrice de covaraince initiale
%
% les explications du roles des facteurs proviennent de [Salgado88]
%
% entrées:
% - alpha,lambda,betha,delta : facteurs d'oubli et de reinitialisation
% - P0 : matrice de covariance de l'erreur initiale
% sorties:
% - foubli :contient alpha,lambda,betha,delta,P0 si les conditions sont
% respectées, sinon effectue une erreur générale avec la première
% condition non respectée
%
function [foubli]=factverif(alpha,lambda,betha,delta,P0)

%avec facteur d'oubli (d'apres travail doctorat Pierre)
%alpha=0.5+0.05;
%lambda=0.9-0.05;

```

```

%beta=0.01*4;
%delta=0.004/2;
%P0=eye(2)*4;

gamma=(1-lambda)/lambda; %pas de confli avec le gamma représentant
l'inclinaison du repère
nu=gamma/(2*delta)*(1+sqrt(1+(4*beta*delta)/gamma^2));
sigma=(alpha-gamma)/(2*delta)*(-1+sqrt(1+(4*beta*delta)/(alpha-
gamma)^2));

if( 0<gamma & gamma<alpha & alpha<1 )
;
else disp('condition 1: non'), error(0)
end

if( (gamma-alpha)^2+4*beta*delta < (1-alpha)^2 )
;
else disp('condition 2: non'), error(0)
end

if( beta>(delta*1e-01) & (delta*1e-01)>0 )
;
else disp('condition 3: non'), error(0)
end

if( sigma*eye(2)<=P0 & P0<=nu*eye(2) )
;
else disp('condition 4: non'), error(0)
end

disp('conditions 1 à 4 sur les facteurs vérifiées')

foubli=[alpha,beta,lambda,delta];

```

C.3.8. Sous-programme : *modgeo.m*

```

% programme : modgeo.m
%
% appel: [xat1,yat1,xat2,yat2,xp,yp,xdmur1,ydmur1,xdmur2,ydmur2]=
% modgeo(x0,y0,x1,y1,x2,y2,gamma,trajet,b,reponse)
%
% But: calcul des paramètres de façon directe (modèle géométrique avec
% bruit)
%
% entrées :
% - x0,y0 : coordonnées en (x,y) du milieu des 2 transducteurs
% - x1,y1 : coordonnées en (x,y) du transducteur 1
% - x2,y2 : coordonnées en (x,y) du transducteur 2
% - b : vecteur de mesure bruitée
% - gamma : inclinaison du mur
% - trajet : nombre d'aller et de retour pour chaque inclinaison
% - reponse: nombre de mur (inclinaison)
% sorties :

```



```

% - xati,yati : coordonnées entre le centre des 2 transducteurs et le
% transducteur i dans le repère fixe
% - xdmuri,ydmuri : coordonnées de la distance détectée par le
% transducteur i dans le repère fixe
% - xp,yp      : coordonnées du centre des 2 transducteurs dans le repère
% fixe
%
function
[xat1,yat1,xat2,yat2,xp,yp,xdmur1,ydmur1,xdmur2,ydmur2]=modgeo(x0,y0,x1,y
1,x2,y2,gamma,trajet,b,reponse);
%save modgeo
%load modgeo

xat10=x1-x0;
yat10=y1-y0;
xat20=x2-x0;
yat20=y2-y0;

omega=atan2(y0,x0);
domega=sqrt(x0.^2+y0.^2);
xp10=domega.*cos(omega+gamma(1).*ones(1,length(x0)));
yp10=domega.*sin(omega+gamma(1).*ones(1,length(x0)));

xp1=[];xp2=[];xp3=[];
yp1=[];yp2=[];yp3=[];
xat11=[];xat12=[];xat13=[];
yat11=[];yat12=[];yat13=[];
xat21=[];xat22=[];xat23=[];
yat21=[];yat22=[];yat23=[];
xdmur11=[];xdmur12=[];xdmur13=[];
ydmur11=[];ydmur12=[];ydmur13=[];
xdmur21=[];xdmur22=[];xdmur23=[];
ydmur21=[];ydmur22=[];ydmur23=[];

for(ii=1:round(trajet(1)))
    xdmur11=b(1,2:round(trajet(1))*24+1)*cos(gamma(1));    %on ajoute 1 à
cause des premieres qui sont nulles
    ydmur11=b(1,2:round(trajet(1))*24+1)*sin(gamma(1));
    xdmur21=b(2,2:round(trajet(1))*24+1)*cos(gamma(1));
    ydmur21=b(2,2:round(trajet(1))*24+1)*sin(gamma(1));

    if((-1)^ii==(-1))
        %impaire%
        xp1=[xp1 xp10];
        yp1=[yp1 yp10];
        xat11=[xat11 xat10*cos(gamma(1))-yat10*sin(gamma(1))];
        yat11=[yat11 xat10*sin(gamma(1))+yat10*cos(gamma(1))];
        xat21=[xat21 xat20*cos(gamma(1))-yat20*sin(gamma(1))];
        yat21=[yat21 xat20*sin(gamma(1))+yat20*cos(gamma(1))];
    else
        %paire%
        xp1=[xp1 fliplr(xp10)];
        yp1=[yp1 fliplr(yp10)];
        xat11=[xat11 fliplr(xat10*cos(gamma(1))-yat10*sin(gamma(1)))];
        yat11=[yat11 fliplr(xat10*sin(gamma(1))+yat10*cos(gamma(1)))];
        xat21=[xat21 fliplr(xat20*cos(gamma(1))-yat20*sin(gamma(1)))];
        yat21=[yat21 fliplr(xat20*sin(gamma(1))+yat20*cos(gamma(1)))];
    end
end

```

```

end
end

if((reponse==2)|(reponse==3))

    xp20=domega.*cos(omega+gamma(2).*ones(1,length(x0)));
    yp20=domega.*sin(omega+gamma(2).*ones(1,length(x0)));

    xdmur12=b(1,(trajet(1)*24+2):(trajet(1)+trajet(2))*24+1)*cos(gamma(2));
    ydmur12=b(1,(trajet(1)*24+2):(trajet(1)+trajet(2))*24+1)*sin(gamma(2));
    xdmur22=b(2,(trajet(1)*24+2):(trajet(1)+trajet(2))*24+1)*cos(gamma(2));
    ydmur22=b(2,(trajet(1)*24+2):(trajet(1)+trajet(2))*24+1)*sin(gamma(2));

    for(ii=round(trajet(1))+1:round(trajet(2))+round(trajet(1)))
        if((-1)^ii==(-1))
            %impaire%
            xp2=[xp2 xp20];
            yp2=[yp2 yp20];
            xat12=[xat12 xat10*cos(gamma(2))-yat10*sin(gamma(2))];
            yat12=[yat12 xat10*sin(gamma(2))+yat10*cos(gamma(2))];
            xat22=[xat22 xat20*cos(gamma(2))-yat20*sin(gamma(2))];
            yat22=[yat22 xat20*sin(gamma(2))+yat20*cos(gamma(2))];
        else
            %paire%
            xp2=[xp2 fliplr(xp20)];
            yp2=[yp2 fliplr(yp20)];
            xat12=[xat12 fliplr(xat10*cos(gamma(2))-yat10*sin(gamma(2)))];
            yat12=[yat12 fliplr(xat10*sin(gamma(2))+yat10*cos(gamma(2)))];
            xat22=[xat22 fliplr(xat20*cos(gamma(2))-yat20*sin(gamma(2)))];
            yat22=[yat22 fliplr(xat20*sin(gamma(2))+yat20*cos(gamma(2)))];
        end
    end

    if(reponse==3)
        xp30=domega.*cos(omega+gamma(3).*ones(1,length(x0)));
        yp30=domega.*sin(omega+gamma(3).*ones(1,length(x0)));

        xdmur13=b(1,(trajet(2)+trajet(1))*24+2:(trajet(1)+trajet(2)+trajet(3))
        *24+1)*cos(gamma(3));
        ydmur13=b(1,(trajet(2)+trajet(1))*24+2:(trajet(1)+trajet(2)+trajet(3))
        *24+1)*sin(gamma(3));
        xdmur23=b(2,(trajet(2)+trajet(1))*24+2:(trajet(1)+trajet(2)+trajet(3))
        *24+1)*cos(gamma(3));
        ydmur23=b(2,(trajet(2)+trajet(1))*24+2:(trajet(1)+trajet(2)+trajet(3))
        *24+1)*sin(gamma(3));

        for(ii=round((trajet(1)+trajet(2))+1):round((trajet(3)+trajet(2)+trajet(1)
        ))))

```

```

    if((-1)^ii==(-1))
        %impair%
        xp3=[xp3 xp30];
        yp3=[yp3 yp30];
        xat13=[xat13 xat10*cos(gamma(3))-yat10*sin(gamma(3))];
        yat13=[yat13 xat10*sin(gamma(3))+yat10*cos(gamma(3))];
        xat23=[xat23 xat20*cos(gamma(3))-yat20*sin(gamma(3))];
        yat23=[yat23 xat20*sin(gamma(3))+yat20*cos(gamma(3))];
    else
        %paire%
        xp3=[xp3 fliplr(xp30)];
        yp3=[yp3 fliplr(yp30)];
        xat13=[xat13 fliplr(xat10*cos(gamma(3))-
yat10*sin(gamma(3)))];
        yat13=[yat13
fliplr(xat10*sin(gamma(3))+yat10*cos(gamma(3)))];
        xat23=[xat23 fliplr(xat20*cos(gamma(3))-
yat20*sin(gamma(3)))];
        yat23=[yat23
fliplr(xat20*sin(gamma(3))+yat20*cos(gamma(3)))];
    end
end
end
end

xp=[0 xp1 xp2 xp3];
yp=[0 yp1 yp2 yp3];
xat1=[0 xat11 xat12 xat13];
yat1=[0 yat11 yat12 yat13];
xat2=[0 xat21 xat22 xat23];
yat2=[0 yat21 yat22 yat23];
xdmur1=[0 xdmur11 xdmur12 xdmur13];    %le zero est déjà compris dans le
calcul de xdmur11
ydmur1=[0 ydmur11 ydmur12 ydmur13];
xdmur2=[0 xdmur21 xdmur22 xdmur23];
ydmur2=[0 ydmur21 ydmur22 ydmur23];

%save modgeo

```

C.3.9. Sous-programme : kalman_song.m

```

% programme: kalman_song
%
% But: calcul l'algorithme de Kalman [Song96] pour l'estimation des
% paramètres ae=(p,q) de l'équation
%       de la tangente (le modèle non lineaire)
%
% Date: 22 septembre 98
%
% Synthaxe: [ae]=kalman_song(nb_ite,b,varb,sd,xp,yp,reponse,trajet,
% gammarep, taille,P0);
% - entrees:
%     nb_ite: nombre d'itération du filtre RLS pour une mesure
%     b:      vecteur des mesures: distance (d1,d2) des transducteurs,

```

```

% angle (thétha) des transducteurs
%   sd:      distance entre les deux transducteurs
%   xp,yp:   coordonnées entre le repère fixe et le repère mobile (les
% transducteurs)
%   reponse: nombre de mur à estimer
%   trajet:   contient le nombre d'aller ou de retour de chaque portion
% de mur
%   gammarep: angle entre le repère fixe et celui mobile
%   P0:       initialisation de la matrice de covariance de l'erreur
%   taille:   nb de points de mesure pour une portion de mur
% - sortie
%   ae:       paramètres (p,q) estimés
%

function
[ae]=kalman_song(nb_ite,b,varb,sd,xp,yp,reponse,trajet,gammarep,taille,P0
);

%save kalman_song
%load kalman_song

P=P0;

aest10=0;      %aest(1,0)
aest20=0;      %aest(2,0)

matPi(1,:)=[P(1) P(3) P(2) P(4)];

ae(:,1)=[aest10;aest20]; %initialisation a la valeur d'origine

if(reponse==1)
    trajet=[trajet 0 0];
end

for(i=2:length(b(1,:)))

    phi(i)=asin(abs(b(1,i)-b(2,i))/sd);
    be(1:3,i)=[b(1,i)*cos(phi(i));b(2,i)*cos(phi(i));b(3,i)];
    eccb=[b(1:3,i)-be(1:3,i)]*[b(1:3,i)-be(1:3,i)]';

    D(i)=2*cos(phi(i));
    dD(i)=-2*(be(1,i)-be(2,i))/(sd*sqrt(sd^2-(be(1,i)-be(2,i))^2));
    mur=1;
    if(i>taille*(trajet(2)+trajet(1))+1)
        mur=3;
    elseif(i>taille*trajet(1)+1)
        mur=2;
    end
    verif(i,:)=[i,mur];
    da(1,i)=(be(1,i)+be(2,i))*cos(b(3,i)+gammarep(mur))/D(i)+xp(i);
    da(2,i)=(be(1,i)+be(2,i))*sin(b(3,i)+gammarep(mur))/D(i)+yp(i);

    aek(:,1)=ae(:,i-1);

    for(k=2:nb_ite+1)

        db(3,k)=-aek(1,k-1)*(da(2,i)-yp(i))+aek(2,k-1)*(da(1,i)-xp(i));

```



```

pcqs(k)=(aek(1,k-1)*cos(b(3,i)+gammarep(mur))+aek(2,k-
1)*sin(b(3,i)+gammarep(mur)))/D(i);
for(x=1:2)
    db(x,k)=pcqs(k)*(1+((-1)^(x))*dD(i)*(be(1,i)+be(2,i))/D(i));
end
C=[da(1,i),da(2,i)];
y(k)=[da(1,i),da(2,i)]*aek(:,k-1)-(aek(1,k-
1)*((b(1,i)+b(2,i))*cos(b(3,i)+gammarep(mur))+xp(i)*D(i))+aek(2,k-
1)*((b(1,i)+b(2,i))*sin(b(3,i)+gammarep(mur))+yp(i)*D(i)))/D(i)+1;
R(k)=[db(1,k),db(2,k),db(3,k)]*eccb*[db(1,k),db(2,k),db(3,k)]';
K(:,k)=P*C'*inv(C*P*C'+R(k));

P=[eye(2)-K(:,k)*C]*P;

%verification du signe des valeurs propres de P
vpP=eig(P);
if((vpP(1)<0)|vpP(2)<0)
    disp('valeur propre négative de P')
    [k,i,vpP']
    pause
end

matP((k-1),:)= [P(1) P(3) P(2) P(4)];
aek(:,k)=aek(:,k-1)+K(:,k)*[y(k)-C*aek(:,k-1)];

end

ae(:,i)=aek(:,k);
Ci(i,:)=C;
Ri(i)=R(k);
Ki(:,i)=K(:,k);
yi(i)=y(k);
matPi(i,:)= [P(1) P(3) P(2) P(4)];
dbi(i,:)=db(:,k)';
pcqsi(i)=pcqs(k);

end

minKi=min(abs(min(abs((Ki(:,2:length(Ki)))))))

```

C.3.10. Sous-programme : kalman_song_salgado.m

```

% programme: kalman_song_salgado
%
% But: calcul l'algorithme de Kalman [Song96] avec adaptation des
% facteurs d'oubli et de reseting
% [Salgado88] pour l'estimation des paramètres ae=(p,q) de
% l'équation de la tangente
% avec le modèle non lineaire.
%
% Date: 22 septembre 98
%
% Synthaxe: [ae]=kalman_song_salgado(nb_ite,b,varb,sd,xp,yp,reponse,
% trajet,taille,gammarep,foubli,P0);

```

```

% - entrees:
%   nb_ite:  nombre d'itération du filtre RLS pour une mesure
%   b:       vecteur des mesures: distance (d1,d2) des transducteurs,
%   angle (théta) des transducteurs
%   sd:      distance entre les deux transducteurs
%   xp,yp:   coordonnées entre le repère fixe et le repère mobile (les
% transducteurs)
%   reponse: nombre de mur à estimer
%   trajet:  contient le nombre d'aller ou de retour de chaque portion
% de mur
%   gammarep: angle entre le repère fixe et celui mobile
%   P0:      initialisation de la matrice de covariance de l'erreur
%   taille:  nb de points de mesure pour une portion de mur
%   foubli:  facteur d'oubli et de resetting exponentiel
% - sortie
%   ae:      paramètres (p,q) estimés
%
function
[ae]=kalman_song_salgado(nb_ite,b,varb,sd,xp,yp,reponse,trajet,taille,gam
marep,foubli,P0);

%save kalman_song_salgado
%load kalman_song_salgado

alpha=foubli(1);
betha=foubli(2);
lambda=foubli(3);
delta=foubli(4);

P=P0;

aest10=0;      %aest(1,0)
aest20=0;      %aest(2,0)

matPi(1,:)= [P(1) P(3) P(2) P(4)];

ae(:,1)=[aest10;aest20]; %initialisation a la valeur d'origine

if(reponse==1)
    trajet=[trajet 0 0];
end

for(i=2:length(b(1,:)))

    phi(i)=asin(abs(b(1,i)-b(2,i))/sd);
    be(1:3,i)=[b(1,i)*cos(phi(i));b(2,i)*cos(phi(i));b(3,i)];
    eccb=[b(1:3,i)-be(1:3,i)]*[b(1:3,i)-be(1:3,i)]';

    D(i)=2*cos(phi(i));
    dD(i)=-2*(be(1,i)-be(2,i))/(sd*sqrt(sd^2-(be(1,i)-be(2,i))^2));
    mur=1;
    if(i>taille*(trajet(2)+trajet(1))+1)
        mur=3;
    elseif(i>taille*trajet(1)+1)
        mur=2;
    end
end

```

```

verif(i,:)=[i,mur];
da(1,i)=(be(1,i)+be(2,i))*cos(b(3,i)+gammarep(mur))/D(i)+xp(i);
da(2,i)=(be(1,i)+be(2,i))*sin(b(3,i)+gammarep(mur))/D(i)+yp(i);

aek(:,1)=ae(:,i-1);

for(k=2:nb_ite+1)

    db(3,k)=-aek(1,k-1)*(da(2,i)-yp(i))+aek(2,k-1)*(da(1,i)-xp(i));
    pcqs(k)=(aek(1,k-1)*cos(b(3,i)+gammarep(mur))+aek(2,k-
1)*sin(b(3,i)+gammarep(mur)))/D(i);
    for(x=1:2)
        db(x,k)=pcqs(k)*(1+((-1)^(x))*dD(i)*(be(1,i)+be(2,i))/D(i));
    end
    C=[da(1,i),da(2,i)];
    y(k)=[da(1,i),da(2,i)]*aek(:,k-1)-(aek(1,k-
1)*((b(1,i)+b(2,i))*cos(b(3,i)+gammarep(mur))+xp(i)*D(i))+aek(2,k-
1)*((b(1,i)+b(2,i))*sin(b(3,i)+gammarep(mur))+yp(i)*D(i)))/D(i)+1;
    R(k)=[db(1,k),db(2,k),db(3,k)]*eccb*[db(1,k),db(2,k),db(3,k)]';
    K(:,k)=alpha*P*C'*inv(C*P*C'+R(k));

    P=[eye(2)/lambda-K(:,k)*C]*P+betha*eye(2)-delta*P^2;

    %verification du signe des valeurs propres de P
    vpP=eig(P);
    if((vpP(1)<0)|vpP(2)<0)
        disp('valeur propre négative de P')
        [k,i,vpP']
        pause
    end

    matP((k-1),:)= [P(1) P(3) P(2) P(4)];
    aek(:,k)=aek(:,k-1)+K(:,k)*[y(k)-C*aek(:,k-1)];

end

ae(:,i)=aek(:,k);
Ci(i,:)=C;
Ri(i)=R(k);
Ki(:,i)=K(:,k);
yi(i)=y(k);
matPi(i,:)= [P(1) P(3) P(2) P(4)];
dbi(i,:)=db(:,k)';
pcqsi(i)=pcqs(k);

end

minKi=min(abs(min(abs((Ki(:,2:length(Ki)))))))

```

C.3.11. Sous-programme : rls_simple

```

% programme: rls_simple
%

```

```

% But: calcul l'algorithme RLS pour l'estimation des paramètres ae=(p,q)
% de l'équation de la tangente
%   le modèle choisie est linéaire
%
%   err(k) = ymes(k) - C(k)'*ae(k)
%
%   avec ymes(k)=1
%         C(k)=[x y] les coordonnées du mur détectés
%         ae(k)=[p q] les coefficients estimés de la tangente
%
%   resolution avec un RLS simple
%
% Date: 22 septembre 98
%
% Synthaxe: [ae]=algo_simple(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,
% P0,taille)
% - entrees:
%   nb_ite: nombre d'itération du filtre RLS pour une mesure
%   b:      vecteur des mesures: distance (d1,d2) des transducteurs,
% angle (théta) des transducteurs
%   sd:     distance entre les deux transducteurs
%   xp,yp:  coordonnées entre le repère fixe et le repère mobile (les
% transducteurs)
%   reponse: nombre de mur à estimer
%   trajet:  contient le nombre d'aller ou de retour de chaque portion
% de mur
%   gammarep: angle entre le repère fixe et celui mobile
%   P0:      initialisation de la matrice de covariance de l'erreur
%   taille:  nb de points de mesure pour une portion de mur
% - sortie
%   ae:      paramètres (p,q) estimés
%
function
[ae]=algo_simple(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille);

%save rls_simple
%load rls_simple

P=P0;

aest10=1e-6;      %différent de 0 à cause de D(i)
aest20=1e-6;

matPi(1,:)=[P(1) P(3) P(2) P(4)];

ae(:,1)=[aest10;aest20]; %initialisation a la valeur d'origine

if(reponse==1)
    trajet=[trajet 0 0];
end

for(i=2:length(b(1,:)))

    phi(i)=asin(abs(b(1,i)-b(2,i))/sd);

    mur=1;

```



```

if(i>taille*(trajet(2)+trajet(1))+1)
    mur=3;
elseif(i>taille*trajet(1)+1)
    mur=2;
end
verif(i,:)=[i,mur];

c1(i)=cos(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+xp(i);
c2(i)=sin(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+yp(i);

aek(:,1)=ae(:,i-1);

for(k=2:nb_ite+1)

    C=[c1(i);c2(i)];
    y(k)=1;
    K(:,k)=P*C*inv(C'*P*C+eye(1));
    P=P-K(:,k)*C'*P;
    %verification du signe des valeurs propres de P
    vpP=eig(P);
    if((vpP(1)<0)|vpP(2)<0)
        disp('valeur propre négative de P')
        [k,i,vpP']
        pause
    end

    matP((k-1),:)= [P(1) P(3) P(2) P(4)];
    aek(:,k)=aek(:,k-1)+K(:,k)*[y(k)-C'*aek(:,k-1)];

end

ae(:,i)=aek(:,k);
Ci(:,i)=C;
Ki(:,i)=K(:,k);
yi(i)=y(k);
matPi(i,:)= [P(1) P(3) P(2) P(4)];

end

minKi=min(abs(min(abs((Ki(:,2:length(Ki)))))))

```

C.3.12. Sous-programme : *rls_lambda*

```

% programme: rls_lambda
%
% But: calcul l'algorithme RLS pour l'estimation des paramètres ae=(p,q)
% de l'équation de la tangente
% le modèle choisie est linéaire
%
% err(k) = ymes(k) - C(k)'*ae(k)
%
% avec ymes(k)=1
% C(k)=[x y] les coordonnées du mur détectés
% ae(k)=[p q] les coefficients estimés de la tangente

```

```

%
%      resolution avec un RLS simple avec un facteur d'oubli
%
% Date: 22 septembre 98
%
% Synthaxe: [ae]=rls_lambda(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,
% taille,lambda)
% - entrees:
%   nb_ite:  nombre d'itération du filtre RLS pour une mesure
%   b:       vecteur des mesures: distance (d1,d2) des transducteurs,
%   angle (théta) des transducteurs
%   sd:      distance entre les deux transducteurs
%   xp,yp:   coordonnées entre le repère fixe et le repère mobile (les
% transducteurs)
%   reponse: nombre de mur à estimer
%   trajet:  contient le nombre d'aller ou de retour de chaque portion
% de mur
%   gammarep: angle entre le repère fixe et celui mobile
%   P0:      initialisation de la matrice de covariance de l'erreur
%   taille:  nb de points de mesure pour une portion de mur
%   lambda:  facteur d'oubli
% - sortie
%   ae:      paramètres (p,q) estimés
%

function
[ae]=rls_lambda(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,lambd
a);

%save rls_lambda
%load rls_lambda

P=P0;

aest10=1e-6;      %différent de 0 à cause de D(i)
aest20=1e-6;

matPi(1,:)= [P(1) P(3) P(2) P(4)];

ae(:,1)=[aest10;aest20]; %initialisation a la valeur d'origine

if(reponse==1)
    trajet=[trajet 0 0];
end

for(i=2:length(b(1,:)))

    phi(i)=asin(abs(b(1,i)-b(2,i))/sd);

    mur=1;
    if(i>taille*(trajet(2)+trajet(1))+1)
        mur=3;
    elseif(i>taille*trajet(1)+1)
        mur=2;
    end
    verif(i,:)=[i,mur];

```

```

c1(i)=cos(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+xp(i);
c2(i)=sin(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+yp(i);

aek(:,1)=ae(:,i-1);

for(k=2:nb_ite+1)

    C=[c1(i);c2(i)];
    y(k)=1;
    K(:,k)=P*C*inv(C'*P*C+eye(1));
    P=P/lambda-K(:,k)*C'*P;
    %verification du signe des valeurs propres de P
    vpP=eig(P);
    if((vpP(1)<0)|vpP(2)<0)
        disp('valeur propre négative de P')
        [k,i,vpP]
        pause
    end

    matP((k-1),:)= [P(1) P(3) P(2) P(4)];
    aek(:,k)=aek(:,k-1)+K(:,k)*[y(k)-C'*aek(:,k-1)];

end

ae(:,i)=aek(:,k);
Ci(:,i)=C;
Ki(:,i)=K(:,k);
yi(i)=y(k);
matPi(i,:)= [P(1) P(3) P(2) P(4)];

end

minKi=min(abs(min(abs((Ki(:,2:length(Ki)))))))

```

C.3.13. Sous-programme : rls_reset.m

```

% programme: rls_reset.m
%
% But: calcul l'algorithme RLS pour l'estimation des paramètres ae=(p,q)
% de l'équation de la tangente
% le modèle choisie est linéaire
%
% err(k) = ymes(k) - C(k) '*ae(k)
%
% avec ymes(k)=1
% C(k)=[x y] les coordonnées du mur détectés
% ae(k)=[p q] les coefficients estimés de la tangente
%
% resolution avec un RLS simple avec reset périodique de la matrice
P
%
% Date: 22 septembre 98
%
% Synthaxe: [ae]=rls_reset(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,

```

```

% taille,treset)
% - entrees:
%     nb_ite:  nombre d'itération du filtre RLS pour une mesure
%     b:       vecteur des mesures: distance (d1,d2) des transducteurs,
% angle (théta) des transducteurs
%     sd:      distance entre les deux transducteurs
%     xp,yp:   coordonnées entre le repère fixe et le repère mobile (les
% transducteurs)
%     reponse: nombre de mur à estimer
%     trajet:  contient le nombre d'aller ou de retour de chaque portion
% de mur
%     gammarep: angle entre le repère fixe et celui mobile
%     P0:      initialisation de la matrice de covariance de l'erreur
%     taille:  nb de points de mesure pour une portion de mur
%     treset:  la période du reset
% - sortie
%     ae:      paramètres (p,q) estimés
%

function
[ae]=rls_reset(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,treset
);

%save rls_reset
%load rls_reset
P=P0;

aest10=0;
aest20=0;

matPi(1,:)=[P(1) P(3) P(2) P(4)];

reset=0;      %changement à la valeur initiale (P0) tout les 10 resets
valreset=[];

ae(:,1)=[aest10;aest20]; %initialisation a la valeur d'origine

if(reponse==1)
    trajet=[trajet 0 0];
end

for(i=2:length(b(1,:)))

    phi(i)=asin(abs(b(1,i)-b(2,i))/sd);

    mur=1;
    if(i>taille*(trajet(2)+trajet(1))+1)
        mur=3;
    elseif(i>taille*trajet(1)+1)
        mur=2;
    end
    verif(i,:)=[i,mur];

    c1(i)=cos(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+xp(i);
    c2(i)=sin(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+yp(i);

    aek(:,1)=ae(:,i-1);

```



```

for(k=2:nb_ite+1)

    if(reset==treset)
        P=P0;
        reset=0;
        valreset=[valreset i];
    else
        reset=reset+1;
    end

    C=[c1(i);c2(i)];
    y(k)=1;
    K(:,k)=P*C*inv(C'*P*C+eye(1));
    P = P - K(:,k)*C'*P ;
    %verification du signe des valeurs propres de P
    vpP=eig(P);
    if((vpP(1)<0)|vpP(2)<0)
        disp('valeur propre négative de P')
        [k,i,vpP']
        pause
    end

    matP((k-1),:)= [P(1) P(3) P(2) P(4)];
    aek(:,k)=aek(:,k-1)+K(:,k)*[y(k)-C'*aek(:,k-1)];

end

ae(:,i)=aek(:,k);
Ci(:,i)=C;
Ki(:,i)=K(:,k);
yi(i)=y(k);
matPi(i,:)= [P(1) P(3) P(2) P(4)];

end

minKi=min(abs(min(abs((Ki(:,2:length(Ki)))))))

```

C.3.14. Sous-programme : *rls_salgado*

```

% programme: rls_salgado
%
% But: calcul l'algorithme RLS pour l'estimation des paramètres ae=(p,q)
% de l'équation de la tangente
% le modèle choisie est linéaire
%
% err(k) = ymes(k) - C(k)'*ae(k)
%
% avec ymes(k)=1
% C(k)=[x y] les coordonnées du mur détectés
% ae(k)=[p q] les coefficients estimés de la tangente
%
% resolution avec un RLS avec facteurs d'oubli et reset exponentiel
%

```

```

% Date: 22 septembre 98
%
% Synthaxe: [ae]=rls_salgado(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,
% P0,taille,foubli)
% - entrees:
%   nb_ite:  nombre d'itération du filtre RLS pour une mesure
%   b:       vecteur des mesures: distance (d1,d2) des transducteurs,
%   angle (théta) des transducteurs
%   sd:      distance entre les deux transducteurs
%   xp,yp:   coordonnées entre le repère fixe et le repère mobile (les
% transducteurs)
%   reponse: nombre de mur à estimer
%   trajet:  contient le nombre d'aller ou de retour de chaque portion
% de mur
%   gammarep: angle entre le repère fixe et celui mobile
%   P0:      initialisation de la matrice de covariance de l'erreur
%   taille:  nb de points de mesure pour une portion de mur
%   foubli:  facteurs d'oubli (cf Salgado88)
% - sortie
%   ae:      paramètres (p,q) estimés
%
function
[ae]=rls_salgado(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,foub
li);

%save rls_salgado
%load rls_salgado

alpha=foubli(1);
betha=foubli(2);
lambda=foubli(3);
delta=foubli(4);

P=P0;

aest10=1e-6;      %différent de 0 à cause de D(i)
aest20=1e-6;

matPi(1,:)= [P(1) P(3) P(2) P(4)];

ae(:,1)=[aest10;aest20]; %initialisation a la valeur d'origine

if(reponse==1)
    trajet=[trajet 0 0];
end

for(i=2:length(b(1,:)))

    phi(i)=asin(abs(b(1,i)-b(2,i))/sd);

    mur=1;
    if(i>taille*(trajet(2)+trajet(1))+1)
        mur=3;
    elseif(i>taille*trajet(1)+1)
        mur=2;
    end
end

```

```

verif(i,:)=[i,mur];

c1(i)=cos(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+xp(i);
c2(i)=sin(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+yp(i);

aek(:,1)=ae(:,i-1);

for(k=2:nb_ite+1)

    C=[c1(i);c2(i)];
    y(k)=1;
    K(:,k)=P*C*inv(C'*P*C+eye(1));
    P=eye(2)*P/lambda-alpha*K(:,k)*C'*P+betha*eye(2)-delta*(P^2);
    %verification du signe des valeurs propres de P
    vpP=eig(P);
    if((vpP(1)<0)|vpP(2)<0)
        disp('valeur propre négative de P')
        [k,i,vpP']
        pause
    end

    matP((k-1),:)= [P(1) P(3) P(2) P(4)];
    aek(:,k)=aek(:,k-1)+alpha*K(:,k)*[y(k)-C'*aek(:,k-1)];

end

ae(:,i)=aek(:,k);
Ci(:,i)=C;
Ki(:,i)=K(:,k);
yi(i)=y(k);
matPi(i,:)= [P(1) P(3) P(2) P(4)];

end

minKi=min(abs(min(abs((Ki(:,2:length(Ki)))))))

```

C.3.15. Sous-Programme : rls_imcevi.m

```

% programme: rls_imcevi
%
% But: calcul l'algorithme RLS pour l'estimation des paramètres ae=(p,q)
% de l'équation de la tangente
% le modèle choisie est linéaire
%
% err(k) = ymes(k) - C(k)'*ae(k)
%
% avec ymes(k)=1
% C(k)=[x y] les coordonnées du mur détectés
% ae(k)=[p q] les coefficients estimés de la tangente
%
% resolution avec un RLS simple avec reset déclenché en cas de
% variation sup. à 10% des coefficients
% comprenant une immunité au reset et un facteur d'oubli lambda
%

```

```

% Date: 22 septembre 98
%
% Synthaxe: [ae]=rls_IMCEVI(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,
% taille,immu,lambda,seuil,biais)
% - entrees:
%   nb_ite: nombre d'itération du filtre RLS pour une mesure
%   b:      vecteur des mesures: distance (d1,d2) des transducteurs,
%   angle (théta) des transducteurs
%   sd:     distance entre les deux transducteurs
%   xp,yp:  coordonnées entre le repère fixe et le repère mobile (les
%   transducteurs)
%   reponse: nombre de mur à estimer
%   trajet: contient le nombre d'aller ou de retour de chaque portion
%   de mur
%   gammarep: angle entre le repère fixe et celui mobile
%   P0:      initialisation de la matrice de covariance de l'erreur
%   taille:  nb de points de mesure pour une portion de mur
%   immu:    le nombre de mesures minimum pour déclencher le reset
%   lambda:  facteur d'oubli
%   seuil:   rapport de variation pour déclencher le reset
%   biais:   seuil minimale de déclenchement du reset
% - sortie
%   ae:      paramètres (p,q) estimés
%
function
[ae]=rls_IMCEVI(nb_ite,b,sd,xp,yp,reponse,trajet,gammarep,P0,taille,immu,
lambda,seuil,biais);

%save rls_dedie
%load rls_dedie

P=P0;

aest10=1e-6;      %différent de 0 à cause de D(i)
aest20=1e-6;

matPi(1,:)=[P(1) P(3) P(2) P(4)];

treset=0;

ae(:,1)=[aest10;aest20]; %initialisation a la valeur d'origine

if(reponse==1)
    trajet=[trajet 0 0];
end

for(i=2:length(b(1,:)))

    phi(i)=asin(abs(b(1,i)-b(2,i))/sd);

    mur=1;
    if(i>taille*(trajet(2)+trajet(1))+1)
        mur=3;
    elseif(i>taille*trajet(1)+1)
        mur=2;
    end
end

```



```

verif(i,:)=[i,mur];

c1(i)=cos(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+xp(i);
c2(i)=sin(b(3,i)+gammarep(mur))*(b(1,i)+b(2,i))/(2*cos(phi(i)))+yp(i);

aek(:,1)=ae(:,i-1);

for(k=2:nb_ite+1)

    %reset en cas de forte variation
    if(i>3)
        if( (abs(ae(1,i-2)-ae(1,i-1))>abs(seuil*ae(1,i-1))) & ...
            (abs(ae(1,i-2)-ae(1,i-1))>biais) ) | ...
            (abs(ae(2,i-2)-ae(2,i-1))>abs(seuil*ae(2,i-1))) & ...
            (abs(ae(2,i-2)-ae(2,i-1))>biais) ) );

            if(treset(length(treset))+immu<i)
                treset=[treset i];
                P=P0;
                aek(:,1)=[0;0];
            else
                noni=i;
            end

        end
    end

    C=[c1(i);c2(i)];
    y(k)=1;
    K(:,k)=P*C*inv(C'*P*C+eye(1));
    P=(P-K(:,k)*C'*P)/lambda;
    %verification du signe des valeurs propres de P
    vpP=eig(P);
    if((vpP(1)<0)|vpP(2)<0)
        disp('valeur propre négative de P')
        [k,i,vpP']
        pause
    end

    matP((k-1),:)= [P(1) P(3) P(2) P(4)];
    aek(:,k)=aek(:,k-1)+K(:,k)*[y(k)-C'*aek(:,k-1)];

end

ae(:,i)=aek(:,k);
Ci(:,i)=C;
Ki(:,i)=K(:,k);
yi(i)=y(k);
matPi(i,:)= [P(1) P(3) P(2) P(4)];

end

minKi=min(abs(min(abs((Ki(:,2:length(Ki)))))))

```

Annexe D :

Compléments sur la schématique

D.1. Étage amplificateur

Cette partie donne la description du schéma d'amplification en courant d'une sortie TTL du micro-contrôleur (HSO.0) pour le signal d'excitation XMIT sur la carte de mesure de chez Polaroid.

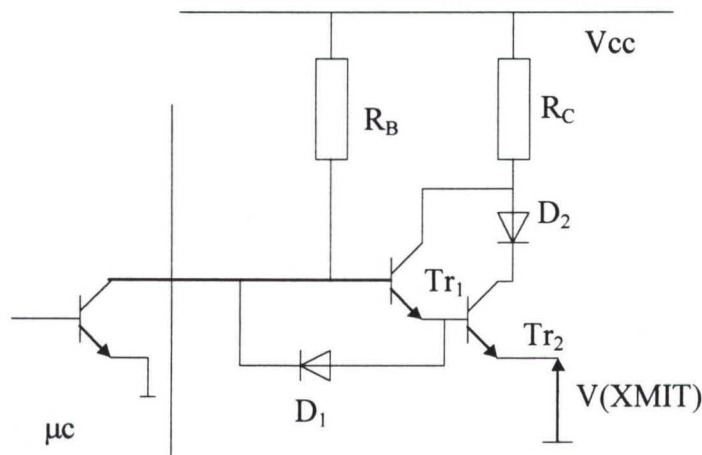


Figure D-1 : Interface micro-contrôleur avec le bloc excitation

On désire avoir $I(XMIT) = 160 \text{ mA}$ et $V(XMIT) = 1.1 \text{ V}$ qui sont appelés respectivement I_{E2} et V . La notation avec l'indice 1 ou 2 se réfère respectivement au transistor 1 ou 2, V_s est la tension de seuil de la diode.

$$I_{B2} = I_{C1} + I_{B1} = (\beta_1 + 1) * I_{B1}$$

$$I_{E2} = I_{C2} + I_{B2} = (\beta_2 + 1) * I_{B2} = (\beta_2 + 1) * (\beta_1 + 1) * I_{B1}$$

$$I_{B1} = I_{E2} / ((\beta_2 + 1) * (\beta_1 + 1))$$

$$V_{cc} = R_B * I_{B1} + V_{BEsat1} + V_{BEsat2} + V$$

$$\Rightarrow R_B = (V_{cc} - V_{BEsat1} - V_{BEsat2} - V) / I_{B1}$$

$$V_{cc} = R_C \cdot I_{C2} + V_s + V_{CEsat2} + V$$

$$\Rightarrow R_C = (V_{cc} - V_s - V_{CEsat2} - V) / (I_{B1} \cdot (\beta_1 \cdot \beta_2 + \beta_2))$$

Il faudra vérifier si le courant entrant dans le micro-contrôleur lorsque les transistors sont bloqués, est inférieur à 5 mA. On prendra le cas extrême avec $5 \text{ mA} \geq (V_{cc} / R_B)$.

Les transistors choisis devront être suffisamment rapides pour ne pas modifier la forme carrée de l'onde excitatrice. On choisira les mêmes transistors, il s'agit du 2N5681.

Le rôle de la diode placée au collecteur de Tr_2 est de représenter une différence de potentiel afin d'avoir l'égalité suivante $V_{BEsat2} + V_{CEsat1} = V_s + V_{CEsat2}$, la diode devra être en plus rapide. L'autre diode (D_1) effectue un drain de courant afin d'absorber le courant emmagasiné au moment de la commutation, ce qui permettra d'obtenir des signaux avec une décharge plus rapide. Là encore, elle devra posséder une rapidité suffisante. Les deux diodes sont identiques, il s'agit de la MUR160 qui possède un temps de recouvrement d'environ 50 ns.

On peut maintenant effectuer l'application numérique: les résistances trouvées seront prises dans la série E12.

$$\Rightarrow R_B = 1.8 \text{ k}\Omega \text{ et } R_C = 22 \text{ }\Omega.$$

D.2. gain variable

Cette partie donne le schéma du gain variable à l'aide des interrupteurs analogiques et du réseau $R/2R$. Les interrupteurs devront être des SPDT, c'est à dire qu'il possède un signal de commande et la sortie peut prendre une des 2 entrées.

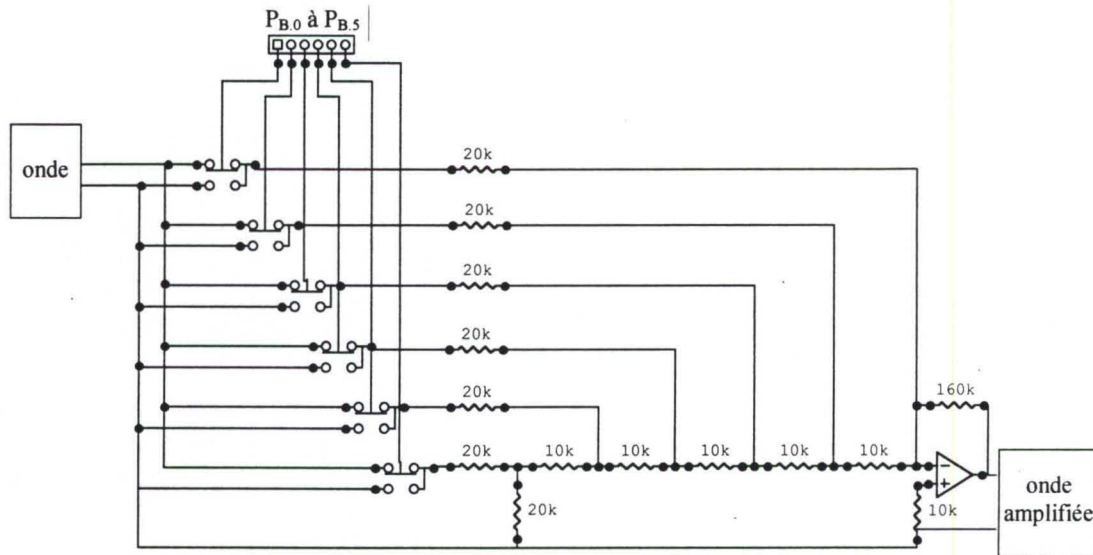


Figure D-2 : Gain variable par réseau R/2R

Dans le montage, le signal reçu est modélisé par une onde sinusoïdale qui devra avoir une valeur moyenne nulle. La commande des interrupteurs se fait de façon logique provenant du micro-contrôleur. Pour obtenir comme plus petit gain 0.25, il faut que le rapport entre la résistance de contre réaction et la puissance de 2 du nombre d'interrupteur soit égal à 0.25. La résistance de contre réaction est 16 fois la résistance de base (10k) car nous utilisons 6 interrupteurs $16/2^6 = 0.25$.

Si on utilise des circuits unipolaires du type CMOS, il est important d'avoir le signal de référence qui soit à une tension fixe et le signal utile soit une tension variable de valeur moyenne la tension de référence.

Il faudra que l'écho ayant l'amplitude la plus grande corresponde au maximum de la plage de mesure du convertisseur A/N. Ici, le montage est polarisé à $0.5 \cdot V_{cc}$ afin d'avoir une amplitude maximale crête à crête de V_{cc} (ajustement avec un gain). La plage du convertisseur va être de 0 à 5 Volt.

Il existe des circuits linéaires qui possèdent le même type de structure interne que celle du gain variable proposée, il s'agit des convertisseurs numériques analogiques à structure R/2R non inversé comme le AD7224 de chez Analog Devices ou l'équivalent chez Maxim

: le PM7224. Cependant, ce type de circuit offre une plage d'entrée limitée, $\pm 1,2$ Volt pour le AD7224. Dans les réalisations effectuées, les deux possibilités ont été testées, et l'utilisation du AD7224 est doublement économique, en place et en prix et sa structure intégrée limite les effets du bruit. Mais la structure interne des interrupteurs analogiques n'est pas entièrement bidirectionnelle, celle-ci pose des problèmes. Ce circuit n'est pas retenu.

Le début de l'étendue de mesure est fixé par la temporisation effectuée de façon logicielle par le micro-contrôleur, ce délai est de 0.902 ms, c'est à dire pour un mur distant de 20 cm.

Annexe E :

Description du micro-contrôleur

80C196KD20

E.1. Les principales caractéristiques du micro-contrôleur

Ce micro-contrôleur fait partie de la famille 80C196, il a une architecture 16 bits.

Circuit compact en 68 ou 80 pattes (entrées ou sorties pouvant être multiplexées)

Fréquence du quartz allant jusqu'à 20 MHz

1000 octets de mémoire RAM

32 ko de ROM écrivable une seule fois (OTPOM)

Sur le 83C196, l'OTPROM est programmée en usine, l'utilisateur le programme sur le 87C196

Une architecture registre à registre, pas d'accumulateur

28 sources d'interruption, 16 vecteurs d'interruption

Un serveur périphérique des transactions (PTS)

Des modes d'économie d'énergie

5 ports d'entrées/sorties de 8 bits, avec une configuration dynamique de la largeur de port (8/16 bits)

Un port série entièrement duplex, protocole de bus HOLD/HOLDA

Des entrées/sorties à haute vitesse (HSI/HSO), 2 HSI, 4 HSO et 2 à se partager.

Un chien de garde (*Watchdog*)

2 Timers 16 bits

Des compteurs (décompteurs) 16 bits avec capture

3 sorties à largeur d'impulsions modulées (PWM)

4 Timers logiciels

8 entrées analogiques de conversion A/D, avec conversion 8 ou 10 bits

E.2. Composition interne

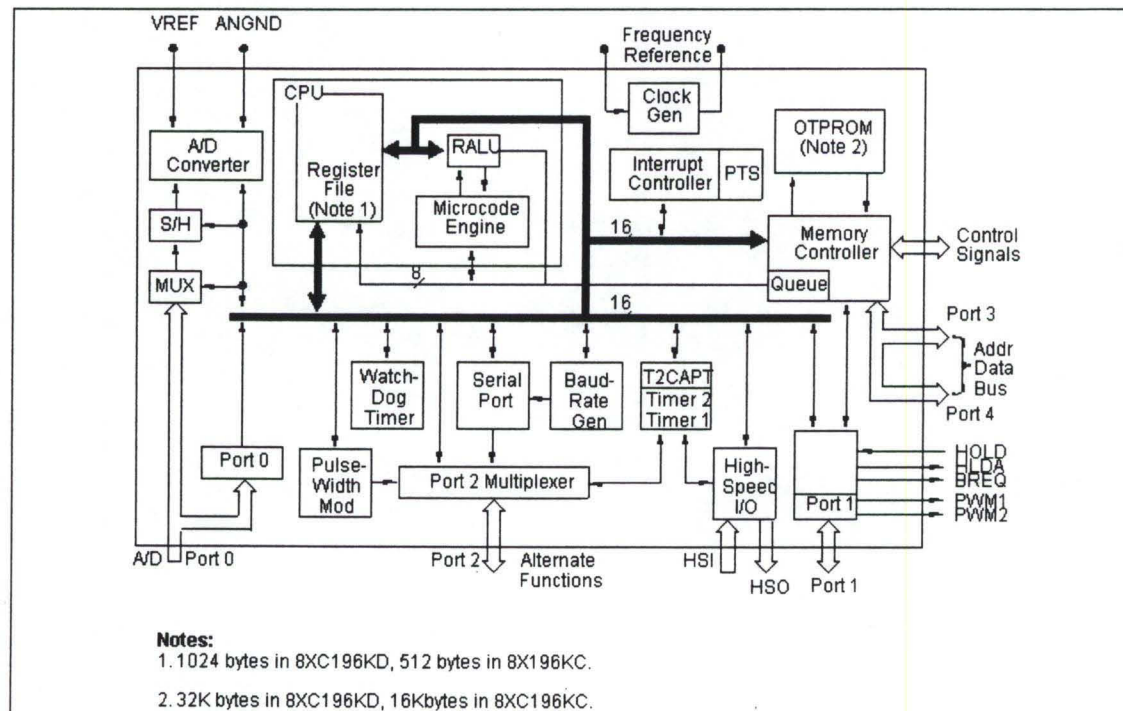


Figure E-1 : Schéma bloc interne du 8XC196KD [Intel92]

Composition interne du processeur (CPU + contrôleur de mémoire)

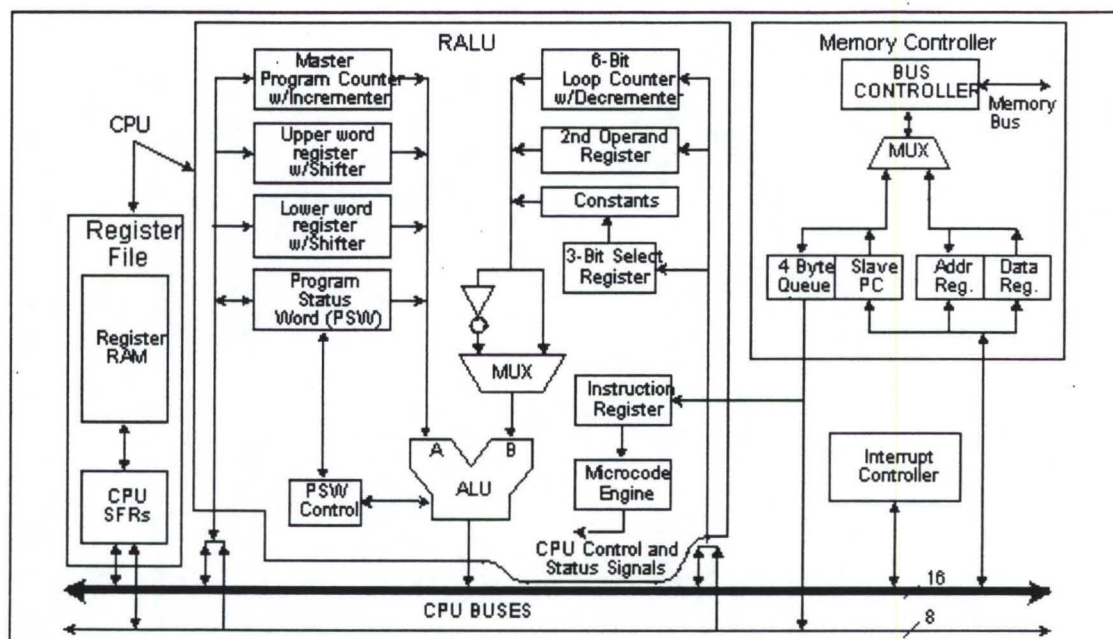


Figure E-2 : Schéma bloc du cœur du 8XC196KD [Intel92]

E.3. Les principales fonctions

E.3.1. Les événements internes

Ils sont de 2 ordres : ceux qui génèrent une interruption ou ceux qui modifient uniquement les drapeaux internes (registres IOSi et IOCi)

E.3.2. Les périphériques internes

Les ports d'entrées et de sorties

Ils sont au nombre 5 ports d'entrées et de sorties (I/O) 8 bits.

Le port 0 est un port en entrée pouvant servir au mode de programmation de l'EPROM ou pour le convertisseur analogique numérique.

Le port 1 est un port bidirectionnel multiplé avec les signaux de contrôle du bus (P1.5 à P1.7) et les 2 sorties PWM.

Le port 2 est un port multifonctionnel qui comprend les signaux pour *Input Capture*, le protocole série et la programmation des ports 3 et 4.

Les port 3 et 4 forment un port 16 bits qui est multiplexé entre le bus de données et le bus d'adresses.

Le port série

Le port série peut être configuré en mode de transmission synchrone et asynchrone, le micro-contrôleur possède en interne un UART.

Les Séquenceurs

Le micro-contrôleur possède 2 séquenceurs matériels (Timer1 et Timer2) et 4 compteurs logiciels (SWT0 à SWT3).

Le Timer1 est un compteur ascendant automatique (*free running*) qui s'incrémente tous les 8 états internes.

Le Timer2 est un compteur programmable (ascendant ou descendant). Son horloge peut être interne (tous les états en mode rapide ou tous les 8 états en mode normal) ou externe. Sa principale fonction est de capturer les événements.

Les compteurs logiciels sont déclenchés par le HSO.

Le HSI

Le HSI possède 2 entrées propres et 2 autres à partager avec le HSO. Son principal rôle est de lire des événements externes (*input capture*).

Le HSO

Le HSO possède 6 sorties comprenant les 2 en partage avec le HSI. Sa référence temporelle peut être aussi bien le Timer1 que le Timer2. Il peut être utilisé avec le CAM qui est un

registre qui contient jusqu'à 8 événements à produire. Il génère les événements internes avec les compteurs logiciels, la remise à zéro du Timer2 ou externes comme le déclenchement du convertisseur A/N, la configuration des sorties du HSO.

Le convertisseur A/N

Le convertisseur analogique numérique est un convertisseur à approximations successives qui donne un résultat de conversion en 8 ou 10 bits. Il peut être configuré à l'aide du HSO ou de la table AD_Scan (événement prédéfini). Les temps de conversion et d'échantillonnage peuvent être déterminés par l'utilisateur selon ces caractéristiques.

Le rapport cyclique programmable

Le micro-contrôleur possède 3 rapports cycliques programmables ou PWM, ce rapport cyclique peut être configuré pour une période de 256 ou de 512 états.

Le chien de garde

Le chien de garde ou *Watchdog* est un compteur indépendant qui vérifie en fond de tâche si le programme n'est pas arrêté anormalement. Si une anomalie est présente, il réinitialise le micro-contrôleur.

Annexe F :

Programme assembleur du micro-contrôleur

La partie la plus importante du programme correspond au choix est l'aiguillage des valeurs échantillonnées en vue de garder ces dernières pour le calcul de l'intercorrélation.

Dans un premier temps, on effectue une acquisition périodique jusqu'à trouver les échantillons qui répondent aux conditions suivantes :

Le temps de l'acquisition est compris entre un temps programmable correspondant à la fin du cycle oscillatoire du transducteur à l'émission et celui correspondant à la portée maximale fixée.

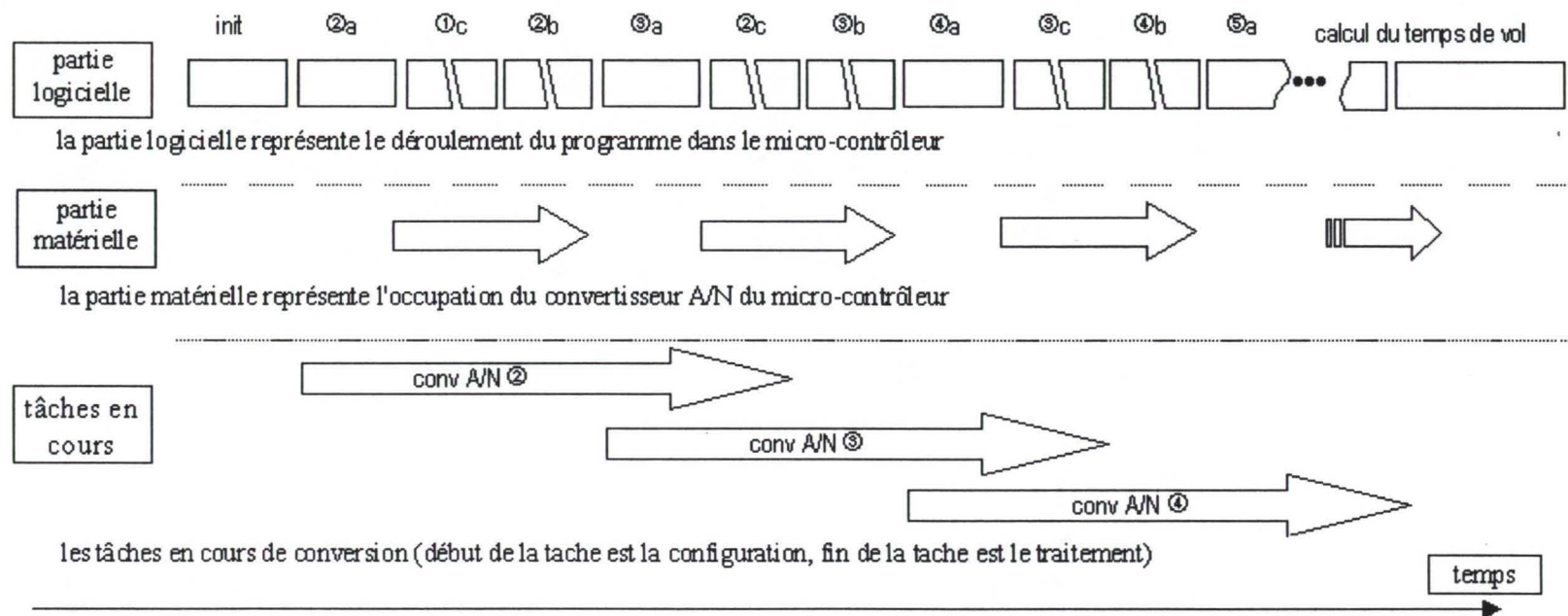
L'amplitude de l'échantillon actuel doit être supérieure au niveau de bruit.

L'amplitude de l'échantillon actuel doit être supérieure à l'amplitude de l'échantillon de retard 2.

Une fois toutes les conditions respectées, le micro-contrôleur emmagasine en mémoire les n_{acq} échantillons à partir de l'échantillon de retard 2. La phase de l'acquisition terminée, le programme effectue un calcul hors ligne de l'intercorrélation avec la recherche de la position du maximum de l'intercorrélation.

La figure suivante (F-1) montre le parallélisme effectué dans le programme de façon que l'acquisition soit la plus rapide. Ce parallélisme est obtenu en effectuant en même temps des tâches matérielles et logicielles au micro-contrôleur.

Figure F-1 : Parallélisme dans l'acquisition



légende:


init: initialisation de la routine, interdiction IT, force le résultat de conversion pour le cycle ① à 0.


a: configuration de la conversion.

b: attente de la fin de la conversion matérielle.

c: traitement de la conversion (valeur absolue, aiguillage du résultat, placement en mémoire ou dans le tampon).

calcul du temps de vol: calcul de l'intercorrélation, détermination du temps de vol.

 temps de l'action variable (selon la mesure)

 temps de l'action fixe

Les organigrammes (F-2 et F-3) donnent une description du programme d'acquisition des échantillons. Les deux derniers (F-4 et F-5) montrent le déroulement du calcul de l'intercorrélation.

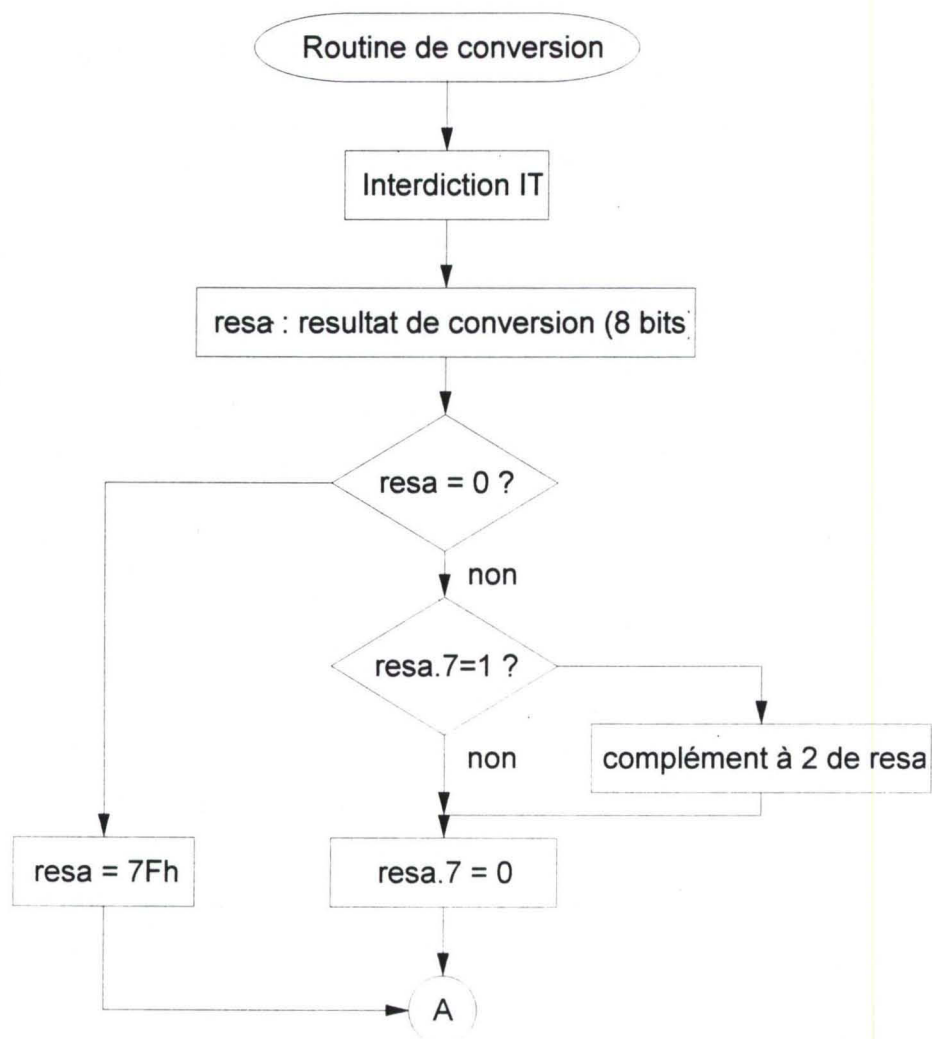


Figure F-2 : Organigramme d'acquisition (1/2)

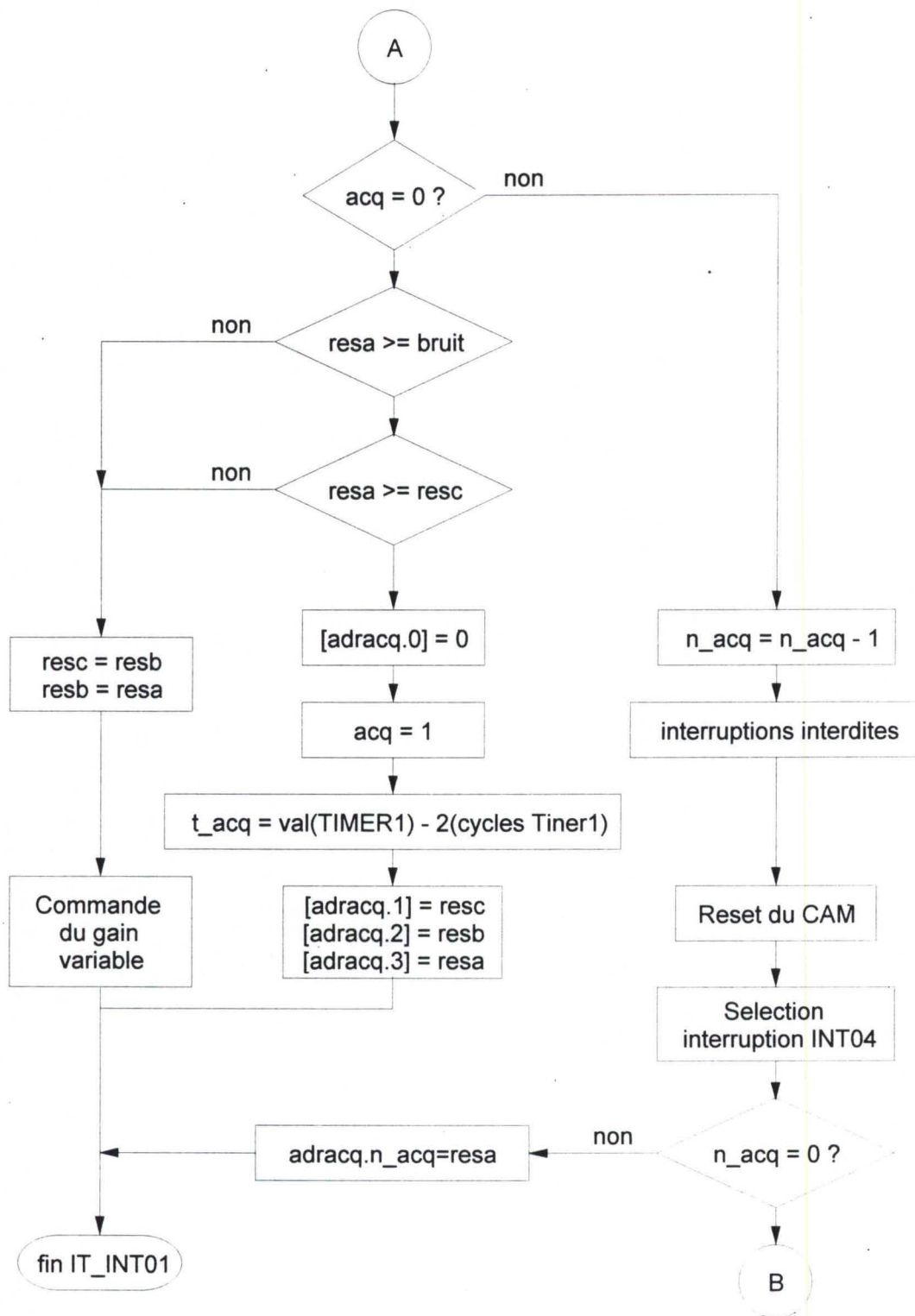


Figure F-3 : Organigramme d'acquisition (2/2)

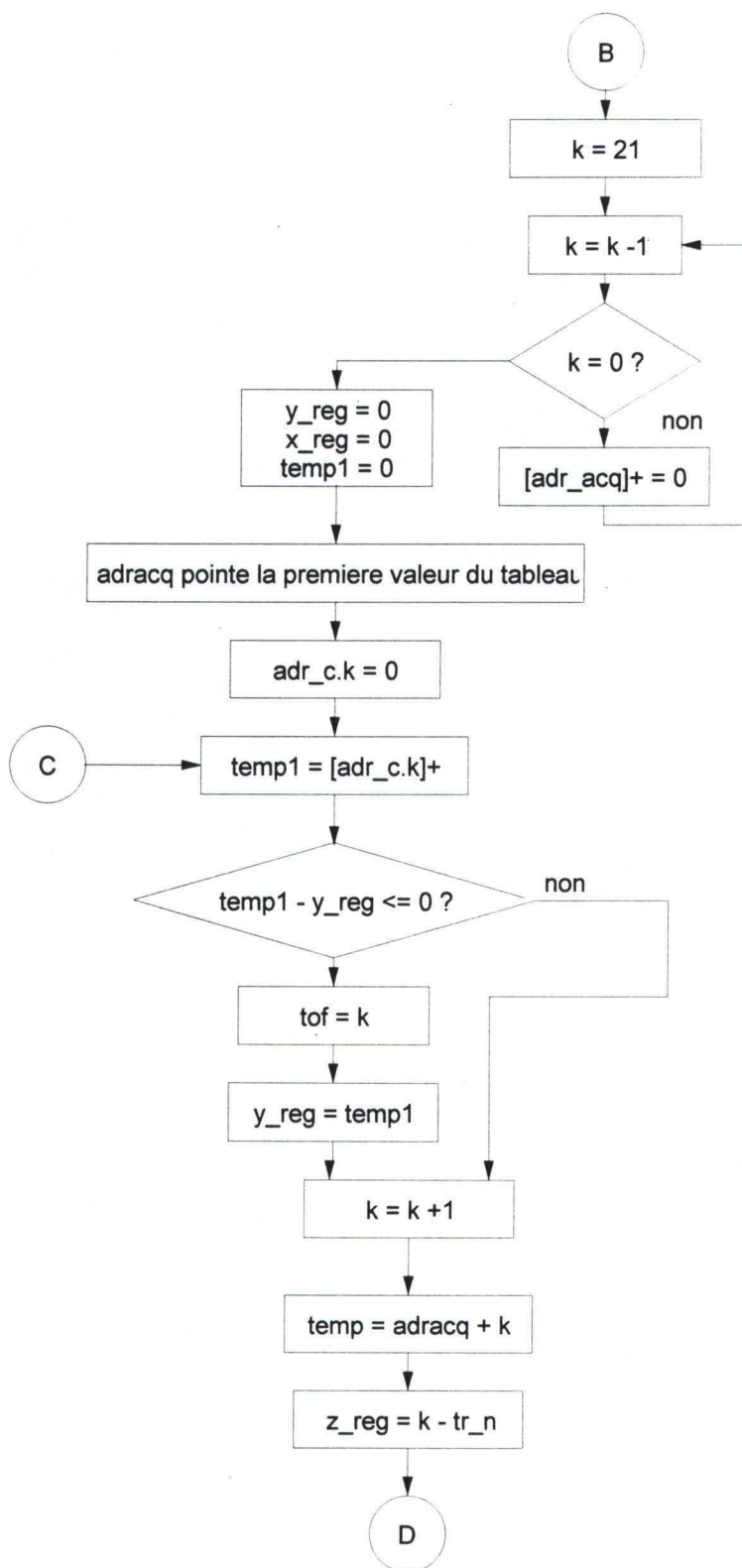


Figure F-4 : Organigramme pour le calcul du temps de vol (1/2)

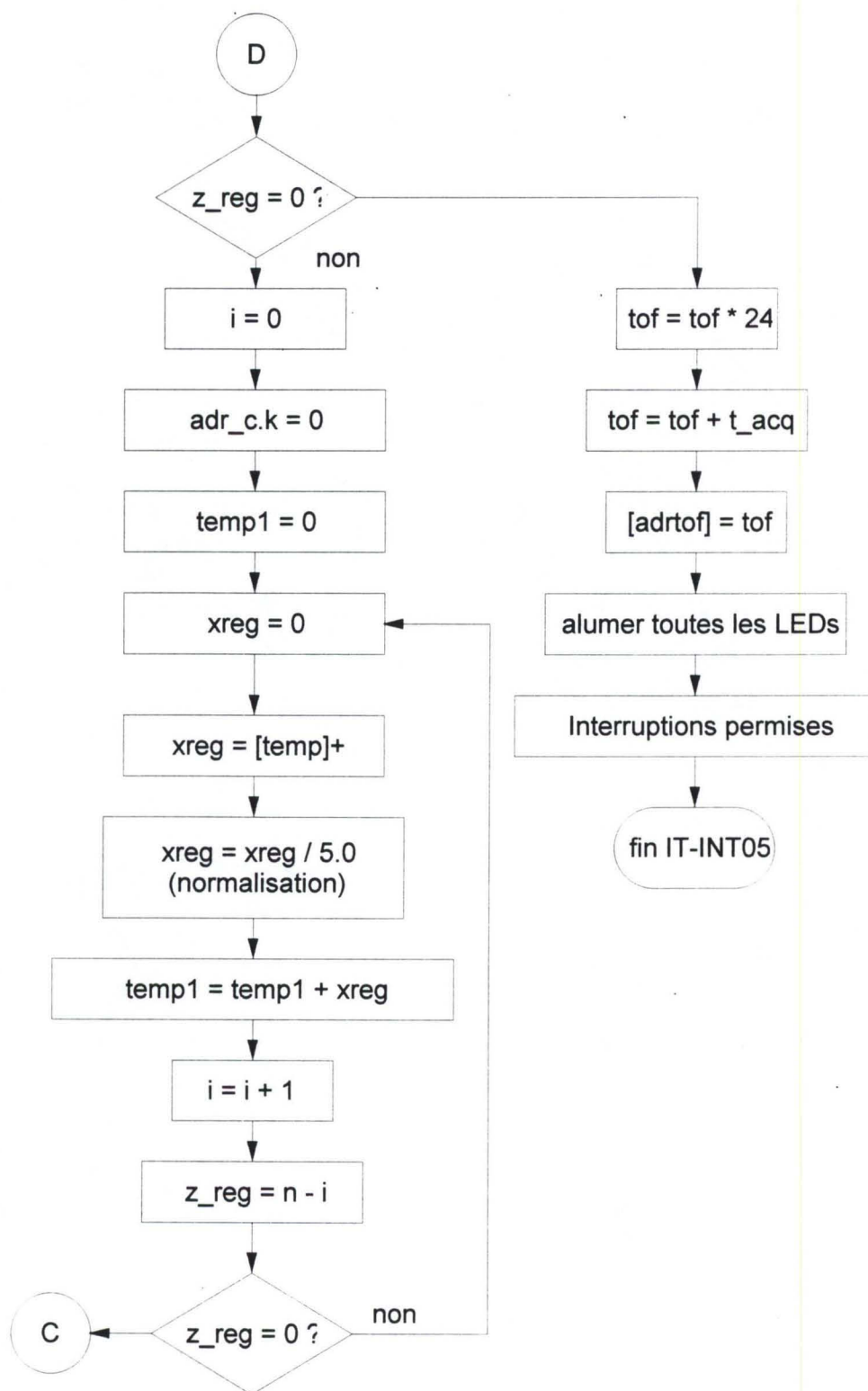


Figure F-5 : Organigramme pour le calcul du temps de vol (2/2)

Annexe G :

Programme assembleur du module de mesure

Le programme qui gère le micro-contrôleur dans le module de mesure de distance est présenté ci-dessous. Ce programme est en assembleur Intel 80x196.

\$debug

```
;*****
;* Programme pour emuler la carte de mesure du circuit ultrasonore *
;*
;* version finale avec gain variable
;*
;* novembre 1998 (emission a 49.5 kHz)
;*****
```

```
;*****
;*
;* Code registre
;*
;*****
```

```
r0 equ 00h:word
zero_reg equ 00h:word
ad_command equ 02h:byte
ad_result equ 02h:word
ad_result_lo equ 02h:byte
ad_result_hi equ 03h:byte
ad_time equ 03h:byte
hsi_mode equ 03h:byte
hso_time equ 04h:word
hsi_time equ 04h:word
hso_command equ 06h:byte
int_mask equ 08h:byte
timer1 equ 0ah:word
ioc2 equ 0bh:byte
ioport1 equ 0fh:byte
int_mask1 equ 13h:byte
wsr equ 14h:byte
ioc0 equ 15h:byte
ios0 equ 15h:byte
ioc1 equ 16h:byte
ios1 equ 16h:byte
```

```
sp          equ    18h:word
```

```
;Declaration des constantes
```

```
rseg at 01ah
```

```
tof          equ    1ah      :word
x_reg        equ    1ch      :word    ;registre tampon (16 bits)
tmpreg       equ    1eh      :byte
i            equ    1fh      :byte
a_timer1     equ    20h      :word
a_hso_time   equ    22h      :word
adres        equ    24h      :word    ;contient la valeur du registre
ad_result    equ    24h      :word
hsotim       equ    26h      :word    ;contient le temps entre 2 conv
A/D
arret        equ    28h      :word    ;variable pour debbugage (pt
arret)
n_emis       equ    2ah      :byte    ;nombre de pts emis
resa         equ    2bh      :byte    ;resultat(a) de conversion
resb         equ    2ch      :byte    ;resultat(b) de conversion
resc         equ    2dh      :byte    ;resultat(c) de conversion
t_acq        equ    2eh      :word    ;temps du debut mise en memoire
acq          equ    30h      :byte
bruit        equ    31h      :byte
n            equ    32h      :byte
z_reg        equ    33h      :byte
adracq       equ    34h      :word
a_timer      equ    36h      :word
tr_n         equ    38h      :byte
y_reg        equ    39h      :byte    ;registre tampon (8 bits)
k            equ    3ah      :byte    ;variable pour la boucle
d'intercor.
n_acq        equ    3bh      :byte    ;nombre de pts pour l'intercor.
adr_c        equ    3ch      :word    ;taille memoire pour l'intercor.
temp         equ    3eh      :word
templ        equ    40h      :word
gtime        equ    42h      :word    ;variable contenant le numero du
gain variable
```

```
dseg at 0046h
```

```
tabacq: dsb    62            ;taille du tableau acq
(40+20(=0))
tab_c: dsb     42            ;taille tableau tab_c (40)
tabgain:dsb    14            ;taille du tableau de gain
tabtime:dsb    14            ;taille du tableau des temps du
gain
```

```
;*****
;*
;*                               Code segment
;*
;*****
```

```
;Interruption du HSI.0
```

```
cseg at 0d080h
    br    IT_int04
    ret
```

; Interruption du software Timer

```
cseg at 0d0a0h
    br    int_05
    ret
```

```
;*****
;*
;*          Programme principal
;*
;*
;*****
cseg at 0C000h
```

```

    di                                ;pas interruption
    ld    SP,#100h
    ld    adracq,#tabacq              ;adresse memoire de l'acquisition
    ld    adr_c,#tab_c                ;adresse memoire de l'intercor.
    ldb   n_acq,#38d                  ;n_acq=38 (pour l'intercor. 40
pts)
    ldb   acq,#0d                     ;acq=0
    ldb   resa,#07fh                  ;initialisation de resa
    ldb   resc,#07fh                  ;initialisation de resc
    ldb   resb,#07fh                  ;initialisation de resb
    ldb   tr_n,#041d                  ;nb de pts pour calculer
l'intercor.
    ldb   bruit,#008h                 ;niveau de bruit
    ldb   n,#020d
    ldb   i,#00h
    stb   i,[adracq]+
    ld    temp,#tabgain               ;adresse memoire du gain variable
    ldb   acq,#000h
    stb   acq,[temp]+
    ldb   acq,#001h                   ;initialisation du gain variable
    stb   acq,[temp]+                 ;valeur de gain(0) a gain(12)
    ldb   acq,#002h
    stb   acq,[temp]+
    ldb   acq,#004h
    stb   acq,[temp]+
    ldb   acq,#006h
    stb   acq,[temp]+
    ldb   acq,#00ch
    stb   acq,[temp]+
    ldb   acq,#010h
    stb   acq,[temp]+
    ldb   acq,#014h
    stb   acq,[temp]+
    ldb   acq,#018h
    stb   acq,[temp]+
    ldb   acq,#01ch
    stb   acq,[temp]+
    ldb   acq,#022h
    stb   acq,[temp]+
    ldb   acq,#028h
```

```

        stb      acq,[temp]+
        ldb      acq,#02fh
        stb      acq,[temp]+
        ldb      acq,#02fh
        stb      acq,[temp]
        ld       temp,#tabgain      ;temp contient l'adresse de la
1ere
                                         ;valeur du gain variable
        ld       temp1,#tabtime     ;temp1 contient l'adresse du 1er
temps
                                         ;du gain variable

        ld       gtime,#0575h
        st       gtime,[temp1]+
        ld       gtime,#018f7h
        st       gtime,[temp1]+
        ld       gtime,#02647h
        st       gtime,[temp1]+
        ld       gtime,#033a7h
        st       gtime,[temp1]+
        ld       gtime,#040ffh
        st       gtime,[temp1]+
        ld       gtime,#04e57h
        st       gtime,[temp1]+
        ld       gtime,#05bafh
        st       gtime,[temp1]+
        ld       gtime,#06907h
        st       gtime,[temp1]+
        ld       gtime,#083d0h
        st       gtime,[temp1]+
        ld       gtime,#09e99h
        st       gtime,[temp1]+
        ld       gtime,#0b962h
        st       gtime,[temp1]+
        ld       gtime,#0d426h
        st       gtime,[temp1]+
        ld       gtime,#0ffdfh
        st       gtime,[temp1]
        ld       temp1,#tabtime     ;temp1 contient l'adresse du 1er
temps
                                         ;du gain variable
        ld       gtime,[temp1]+     ;gtime contient le 1er temps du
gain variable
        ld       adres,#0008h      ;valeur du registre ad_result
        ld       hsotim,#0028d     ;temps entre 2 conversion A/D
        ld       arret,#0fedfh     ;temps pour arreter la conversion
A/D
        ldb      n_emis,#15d        ;nb de pts emis=15
        ldb      acq,#00h           ;acq=0
        ldb      wsr,#00h           ;Hwindows=0
        ldb      ioc2,#080h         ;reset du CAM + possibilite
d'enregistrement
        ldb      int_mask,#010h     ;IT sur HSI.0
        ei                          ;interruption autorisee
loop:    sjmp     loop              ;attente IT

```

```

;*****
;*

```

```

*
```



```

;*          routine d'interruption int04 (HSI.0)          *
;*          *                                              *
;*****
;Le train de 15 ondes carrees est genere par des instructions
;d'attente pour avoir precisement 102 etats par niveau logique
;c'est a dire une frequence de 49.5kHz (au lieu d'utiliser un
;evenement programme a 8 etats pres).
;Effectue un masquage de l'echo emis en autorisant l'evenement
;avec le timer logiciel au bout de 1,02ms (= objet a 17.5cm)

cseg at 0c200h
IT_int04:
    di                                ;IT interdit
    ldb     wsr,#00h
    ldb     n_emis,#015d
    ldb     wsr,#00fh
    clr     timer1
    ldb     ios0,#001h
    br      train

train:
    djnz    n_emis,debut
    br      fin

debut:
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    nop
    nop
    nop
    ldb     wsr,#00fh
    ldb     ios0,#00h                ;clear hso.0

    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    shr     zero_reg,#2h
    skip    r0
    skip    r0
    skip    r0
    nop
    ldb     ios0,#001h                ;set hso.0
    br      train

fin:
    shr     zero_reg,#2h

```

```

shr      zero_reg,#2h
shr      zero_reg,#2h
shr      zero_reg,#2h
shr      zero_reg,#2h
shr      zero_reg,#2h
shr      zero_reg,#2h
shr      zero_reg,#2h
nop
nop
nop
ldb      ios0,#00h

;* fin de la creation du train de 15 impulsions

shr      zero_reg,#2h

ldb      wsr,#00h
st       timer1,a_timer1

shr      zero_reg,#2h

ldb      wsr,#0fh
ldb      tmpreg,ioc2
orb      tmpreg,#0c0h          ;effectue un reset du CAM
ldb      wsr,#00h
ldb      ioc2,tmpreg

ldb      wsr,#0fh
ldb      tmpreg,int_mask
andb     tmpreg,#0f7h          ;interdit int03 (hso)
orb      tmpreg,#020h          ;autorise int05 (software timer)
ldb      wsr,#00h
ldb      int_mask,tmpreg

ld       a_timer1,#0500h
ldb      hso_command,#018h      ;config. evenement sur software

timer
st       a_timer1,hso_time
ldb      wsr,#00fh
st       hso_time,a_hso_time

ldb      wsr,#00h
ei
ret                      ;retour au programme principale

;*****
;*
;*      routine d'interruption int05 (Timer logiciel)
;*
;*****
;Configuration de l'acquisition automatique
; - config. de la conversion declenchee par hso
; - pas de configuration de int_mask
; - on n'utilise pas IT sur fin conversion A/D
;Generation des commandes du gain variable
;Calcul de l'intercorrelation

```

```
;Recherche de la position du maximum de l'intercorrelation
;Donne le temps de vol en cycles du Timer1
```

```
cseg at 0ca00h
```

```
int_05:
```

```
di
```

```
;* configuration de la prochaine conversion A/N *
```

```
    ldb    wsr,#01h
    ldb    ad_time,#068h
    ldb    wsr,#00h
    ldb    ioc2,#088h          ;AD_Time register + CAM reset
    ldb    ad_command,#010h    ;conv 8 bits avec hso
    shr    zero_reg,#2h
    ldb    hso_command,#00fh   ;debut conv, pas it, pas lock
    add    a_hso_time,hsotim
    ld     hso_time,a_hso_time
    shr    zero_reg,#2h

    ldb    wsr,#00fh
    or     ad_result,adres     ;met a 1 le bit 3 (conversion en
court)
    ldb    wsr,#00h
```

```
;routine qui effectue la valeur absolue
;resa, resb, resc sont configurees dans le prog.princ.
```

```
conv:
```

```
abs:    andb    resa,#0FFh          ;operation ne modifiant pas resa
        jne     no_min              ;branchement si resa!=0
```

```
min:    ldb     resa,#07Fh          ;si resa=0 -> resa=7F
        sjmp    f_abs
```

```
no_min: jbs     resa,07h,posi        ;test du bit7
;branchement si resa inf à 128 (nb negatif)
nega:   negb    resa                ;complement a 2
posi:   andb    resa,#07Fh          ;mise a 0 du bit 7
```

```
;effectue le pretraitement de la derniere conversion
```

```
f_abs:
        addb    acq,#0h              ;equivaut a cmpb
        jne     acq1                ;si acq!=0, branchement a acq1
acq0:   subb    z_reg,resa,bruit      ;z_reg=resa-bruit
        jle     nulacq              ;si resa<=bruit -> nulacq
        subb    z_reg,resa,resc      ;z_reg=resa-resc
        jle     nulacq              ;si resa<=resc -> nulacq
        ldb     acq,#1d              ;acq=1
        sub     t_acq,timer1,#056d   ;t_acq=Timer1-2cycles
        stb     resc,[adracq]+       ;\
        stb     resb,[adracq]+       ;-mise en memoire de
resc,resb,resa
        stb     resa,[adracq]+       ;/
        br      test                 ; recommence A/D
```

```

nulacq: ldb      resc,resb          ;resc <- resb
        ldb      resb,resa         ;resb <- resa

;gain variable

gtof:   ldb      wsr,#00h
        st       timer1,a_timer1
        cmp      gtime,a_timer1
        jh       gain
        cmp      a_timer1,arret    ;verifie l'overflow (ffdf)
        jh       overflow
        ld       gtime,[temp1]+
        inc      temp
gain:   ldb      ioport1,[temp]     ;ecriture sur le port (wsr=0)
        br       test              ;recommance A/D (pour pt arret)

acq1:   djnz     n_acq,memacq       ;n_acq-=1, si n_acq!=0 ->
branchement
        br       inter
memacq: stb      resa,[adracq]+     ;mise en memoire d'un nouveau
point

;boucle d'attente de fin de conversion

test:   nop
        jbs      ad_result,3,test

        ldb      wsr,#00h

;lecture du resultat

        stb      ad_result_hi,resa

;configuration de la nouvelle conversion

        ldb      wsr,#01h
        ldb      ad_time,#068h
        ldb      wsr,#00h
        ldb      ioc2,#008h        ;AD_Time register + CAM reset
        ldb      ad_command,#010h  ;conv 8 bits avec hso
        ldb      hso_command,#00fh ;debut conv, pas it, pas lock
        add      a_hso_time,hsotim
        ld       hso_time,a_hso_time
        ldb      wsr,#00fh
        or       ad_result,adres   ;met a 1 le bit 3 conversion en
court
        ldb      wsr,#00h
        br       conv

overflow:
        ldb      wsr,#00h
        ldb      ioport1,#010000000b ;affichage de la LED du MSB
        br       fin_int_05

;calcul de l'intercorrelation

```



```

inter:  ldb      k,#21h                ;k=21 (il faut ajouter 20 pts a
0)
m0_acq: djnz     k,m1_acq              ;k=k-1, si k!=0 branchement a
m1_acq  br       xcor                  ;sinon branchement a xcor
m1_acq: stb      i,[adracq]+           ;mise a (i=0) le contenu de la
memoire adracq
        br m0_acq

xcor:   ldb      resa,#0d              ;resa=0
        ldb      resc,#0d             ;resc=0, k=0 (au dessus)
        ldb      y_reg,#0d            ;y_reg=0
        ldb      templ,#00h
        ld       adracq,#tabacq
        add      temp,adracq,k         ;temp=tabacq.0
        and      temp,#00ffh          ;temp_HI=00

i_k:    stb      templ,[adr_c]+        ;sauvegarde de la val
intercorrelation pour k
        cmpb     templ,y_reg          ;c(k)-y_reg(precedant max)
        jnh      suite               ;si resultat<=0 , branchement à
suite   ld       tof,k                ;tof=k
        ldb      y_reg,templ          ;

suite:  incb      k                    ;k=k+1

;initialisation de mon pointeur d'adresse a la valeur k (temp)

        add      temp,adracq,k         ;temp=tabacq.k
        and      temp,#00ffh          ;temp_HI=00

        subb     z_reg,k,tr_n          ;z_reg=k-tr_n
        je       f_xcor               ;si k=tr_n -> branchement a
f_xcor  ldb      i,#0h                ;i=0
        stb      i,[adr_c]            ;adr_c.k=0
        stb      i,templ              ;templ=0

i_i:    ld       x_reg,#0000h
        ldb      x_reg,[temp]+
        divub    x_reg,#05d
        addb     templ,x_reg

        incb     i                    ;i=i+1
        subb     z_reg,n,i            ;z_reg=n-i
        je       i_k                 ;si i=n -> branchement a i_i
        br      i_i

;routine d'enregistrement du temps correspondant au max de c
f_xcor: mulub    tof,#24d              ;tof=t(max(c))*24, correspond
(enchantillonnage)                  ;en cycle de TIMER1

```

```

    add    tof,t_acq          ;tof=t(max(c))+t_acq

;routine d'averisement lumineux
    ldb    IOPORT1,#011000000b    ;affichage des 2 LEDs du MSB

fin_int_05:

;configuration pour la prochaine émission
    ld     adr_acq,#tabacq        ;adresse memoire de l'acquisition
    ld     adr_c,#tab_c          ;adresse memoire de l'intercor.
    ldb    n_acq,#38d            ;n_acq=38 (pour l'intercor. 40
pts)
    ldb    acq,#0d               ;acq=0
    ldb    resa,#07fh            ;initialisation de resa
    ldb    resc,#07fh            ;initialisation de resc
    ldb    resb,#07fh            ;initialisation de resb
    ldb    tr_n,#041d            ;nb de pts pour calculer
l'intercor.
    ldb    bruit,#008h           ;niveau de bruit
    ldb    n,#020d
    ldb    i,#00h
    stb    i,[adr_acq]+
    ld     temp1,#tabtime        ;temp1 contient l'adresse du 1er
temps
                                ;du gain variable
    ld     gtime,[temp1]+
gain variable                    ;gtime contient le 1er temps du
    ld     temp,#tabgain        ;temp contient l'adresse de la
lere
                                ;valeur du gain variable
    ld     adres,#0008h          ;valeur du registre ad_result
    ldb    n_emis,#15d           ;nb de pts emis=15
    ldb    wsr,#00h              ;Hwindows=0
    ldb    ioc2,#080h            ;reset du CAM + possibilite
d'enregistrement
    ldb    int_mask,#010h        ;IT sur HSI.0

    ei                            ;interruption autorisee

    ret

end

```

Annexe H :

Documentation sur les composants ultrasonores cités

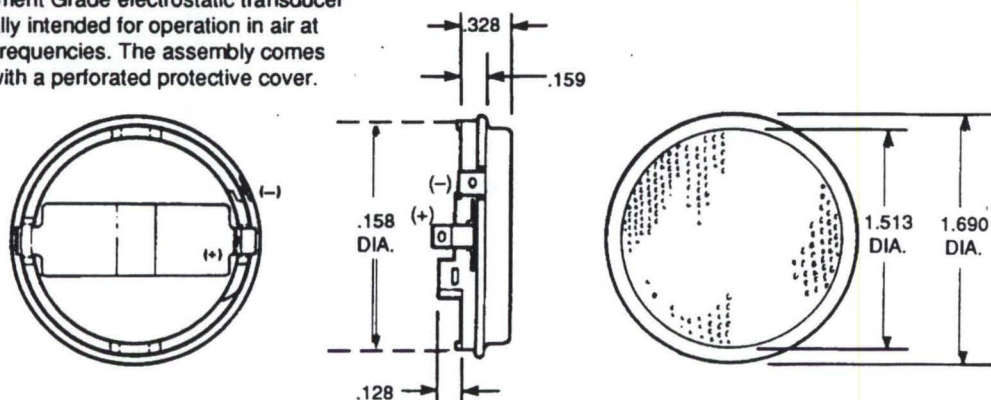
La documentation suivante provient du catalogue : Ultrasonic components de chez Polaroid OEM, 02/1996. Les spécifications techniques de 4 composants sont présentées :

- Le transducteur utilisé, *600 Series Instrument Grade Electrostatic Transducer*, (page H-2).
- Le petit module de mesure dont une petite partie sert pour l'interface entre le module de mesure et le transducteur, *6500 Series Sonar Ranging Module*, (pages H-3 à H-6).
- Le TL851 qui commandait le gain variable et détectait si l'amplitude de l'enveloppe de l'écho avait atteint le seuil, *Digital Chip*, (pages H-7 à H-10).
- Le TL852 qui effectuait le gain variable et l'enveloppe de l'écho, *Analog Chip*, (pages H-11 à H-14).

Technical Specifications for 600 Series Instrument Grade Electrostatic Transducer

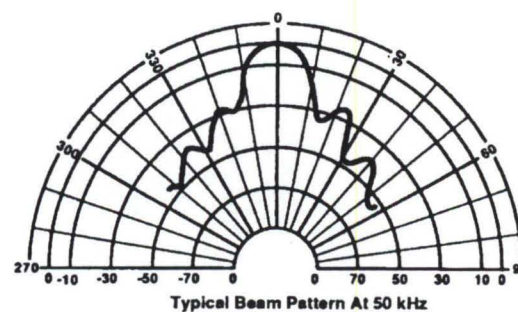
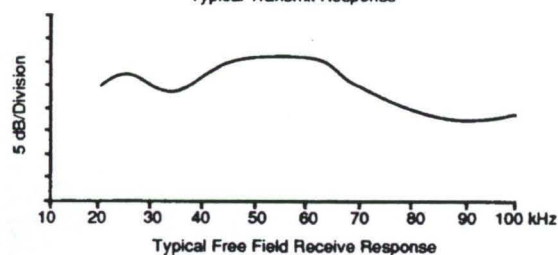
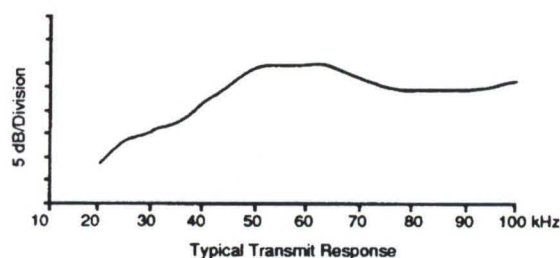
Part# 616341 -Minimum order 10. Increments of 10....Part# 604142-Minimum order 100, increments of 100

This Instrument Grade electrostatic transducer is specifically intended for operation in air at ultrasonic frequencies. The assembly comes complete with a perforated protective cover.



Specifications:

Usable Transmitting Frequency Range	See Graph	Maximum Combined Voltage	400V
Usable Receiving Frequency Range	See Graph	Capacitance at 1 kHz (Typical)	400-500 pf
Beam Pattern	See Graph	150 vdc bias	
Minimum Transmitting Sensitivity at 50 kHz	110 dB	Operating Conditions	
300 vacpk-pk, 150 vdc bias		Temperature	-20°-160°F
(dB re 20 μ Pa at 1 meter)		Relative Humidity	5%-95%
Minimum Receiving Sensitivity at 50 kHz	-42dB	Standard Finish	
(150 vdc bias (dB re 1v/Pa)		Foil	Gold
Suggested DC Bias Voltage	150V	Housing	Flat Black
Suggested AC Driving Voltage (peak)	150V		Cold Roll Steel
Specifications subject to change without notice.			



Note: dB normalized to on-axis response.

Note: Curves are representative only. Individual responses may differ.

Technical Specifications for

6500 Series Sonar Ranging Module

Part# 615077- Minimum order 10, increments of 10

Features:

- Accurate Sonar Ranging from 6" to 35 ft.
- Drives 50 kHz Electrostatic Transducer with No Additional Interface
- Operates from Single Supply
- Accurate Clock Output Provided for External Use
- Selective Echo Exclusion
- TTL-Compatible
- Multiple Measurement Capability
- Uses TI TL851 and Polaroid 614906 Sonar Chips
- Socketed Digital Chip
- Convenient Terminal Connector
- Variable Gain Control Potentiometer

Description:

The 6500 Series is an economical sonar ranging module that can drive all Polaroid electrostatic transducers with no additional Interface. This module, with a simple interface, is able to measure distances from 6" to 35 ft. The typical absolute accuracy is $\pm 1\%$ of the reading over the entire range.

This module has an external blanking input that allows selective echo exclusion for operation on a multiple echo mode. The module is able to differentiate echoes from objects that are only 3" apart. The digitally controlled-gain, variable-bandwidth amplifier minimizes noise and side-lobe detection in sonar applications.

The module has an accurate ceramic resonator-controlled 420 kHz time-base generator. An output based on the 420 kHz time base is provided for external use. The sonar transmit output is 16 cycles at a frequency of 49.4 kHz.

The 6500 Series module operates over a supply voltage range from 4.5 volts to 6.8 volts and is characterized for operation from 0° to 40°C.

Absolute Maximum Ratings:

Voltage from any pin to ground (see Note 1).....	7V
Voltage from any pin except XDOR to Vcc (see Note 1)	-7 to 0.5 V
Operating free-air temperature range	0°C to 40°C
Storage temperature range.....	40°C to 85°C

NOTE 1: The XDOR pin may be driven from -1 volt to 400 volts typical with respect to ground

Specifications subject to change without notice.

Recommended Operating Conditions

	MIN	MAX	UNIT
Supply Voltage, Vcc	4.5	6.8	V
High-level Input Voltage, V_{IH}	2.1		V
Low-level Input Voltage, V_{IL}		0.6	V
ECHO and OSC Output Voltage		6.8	V
Delay Time, Power Up to INIT High	5		ms
Recycle Period	80		ms
Operating Free-air Temperature, T_A	0	40	°C

Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature (Unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input Current	BLNK, BINH, INIT	V _I = 2.1V			1	mA
High-level Output Current, I _{OH}	ECHO, OSC	V _{OH} = 5.5V			100	uA
Low-level Output Voltage, V _{OL}	ECHO, OSC	I _{OL} = 1.6 mA			0.4	V
Transducer Bias Voltage		T _A = 25°C		200		V
Transducer Output Voltage (peak-to-peak)		T _A = 25°C		400		V
No. of Cycles for XDCR Output to Reach 400V		C = 500pF			7	
Internal Blanking Interval				2.38†		ms
Frequency During 16-pulse Transmit Period	OSC output			49.4†		kHz
	XMIT output			49.4†		kHz
Frequency After 16-pulse Transmit Period	OSC output			93.3†		kHz
	XMIT output			0		kHz
Supply Current, I _{cc}	During transmit period				2000	mA
	After transmit period				100	mA

These typical values apply for a 420 kHz ceramic resonator.

Operation With Polaroid Electrostatic Transducer:

There are two basic modes of operation for the 6500 Series sonar ranging module: single-echo mode and multiple-echo mode. The application of power (V_{cc}), the activation of the Initiate (INIT) input, and the resulting transmit output, and the use of the Blanking Inhibit (BINH) input are basically the same for either mode of operation. After applying power (V_{cc}) a minimum of 5 milliseconds must elapse before the INIT input can be taken high. During this time, all internal circuitry is reset and the internal oscillator stabilizes. When INIT is taken high, drive to the Transducer XDCR output occurs. Sixteen pulses at 49.4 kHz with 400volt amplitude will excite the transducer as transmission occurs. At the end of the 16 transmit pulses, a DC bias of 200 volts will remain on the transducer as recommended for optimum operation.

In order to eliminate ringing of the transducer from being detected as a return signal, the Receive (REC) input of the ranging control IC is inhibited by Internal blanking for 2.38 milliseconds after the initiate signal. If a reduced blanking time is desired, then the BINH input can be taken high to end the blanking of the Receive input anytime prior to Internal blanking. This maybe desirable to detect objects closer than 1.33ft. corresponding to 2.38 milliseconds and may be done if transducer damping is sufficient so that ringing is not detected as a return signal.

In the single-echo mode of operation (Figure 1), all that must be done next is to wait for the return of the transmitted signal, traveling at approximately 0.9 milliseconds per foot out and back. The returning signal is amplified and appears as a high-logic-level echo output. The time between INIT going high and the Echo (ECHO) output going high is proportional to the distance of the target from the transducer. If desired, the cycle can now be repeated by returning INIT to a low-logic level and then taking it high when the next transmission is desired.

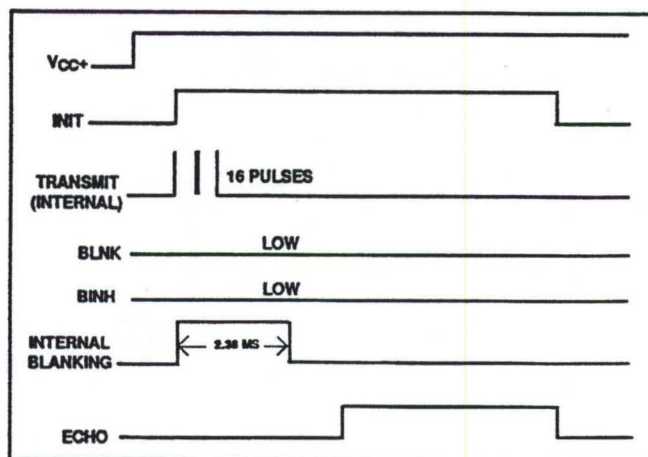
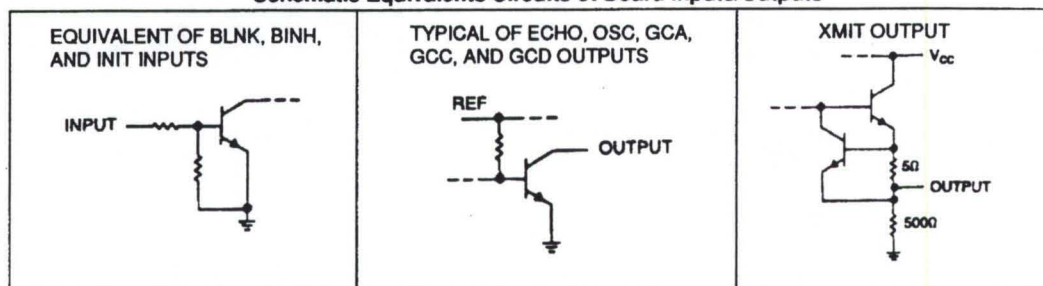


FIGURE 1: EXAMPLE OF A SINGLE-ECHO-MODE CYCLE WITHOUT BLANKING INPUT

Schematic Equivalents Circuits of Board Inputs/Outputs



INPUT/OUTPUT SCHEMATICS NOTE: The echo output is an open collector transistor output and requires a 4.7 k ohm pull up resistor between V_{CC} and the output.

If there is more than one target and multiple echoes will be detected from a single transmission, then the cycle is slightly different (Figure 2). After receiving the first return signal which causes the ECHO output to go high, the Blanking (BLNK) input must be taken high then back low to reset the ECHO output for the next return signal. The blanking signal must be at least 0.44 milliseconds in duration to account for all 16 returning pulses from the first target and allow for internal delay times. This corresponds to the two targets being 3" apart.

During a cycle starting with INIT going high, the receiver amplifier gain is incremented higher at discrete times (Figure 3) since the transmitted signal is attenuated with distance. At approximately 38 milliseconds, the maximum gain is attained. For this reason, sufficient gain may not be available for objects greater than 35 ft. away. Although gain can be increased by varying R1 (Figure 4), there is a limit to which the gain can be increased for reliable module operation. This will vary from application to application. The modules are "kitted" prior to their final test during manufacture. This is necessary because the desired gain distribution is much narrower than the module gain distribution if all were kitted with one value resistor. As kitted, these modules will perform satisfactorily in most applications. As a rule of thumb, the gain can be increased up to a factor of 4, if required, by increasing R1 correspondingly. Gain is directly proportional to R1.

Potentiometer VR1 (Figure 4) provides an interstage gain adjustment for the module. It can be used to trim the overall range of gain set by fix resistor R1.

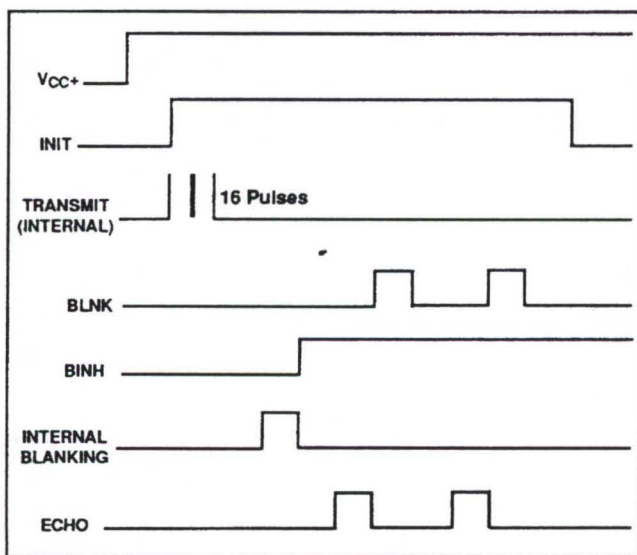


FIGURE 2: EXAMPLE OF A MULTIPLE-ECHO-MODE CYCLE WITH BLANKING INPUT

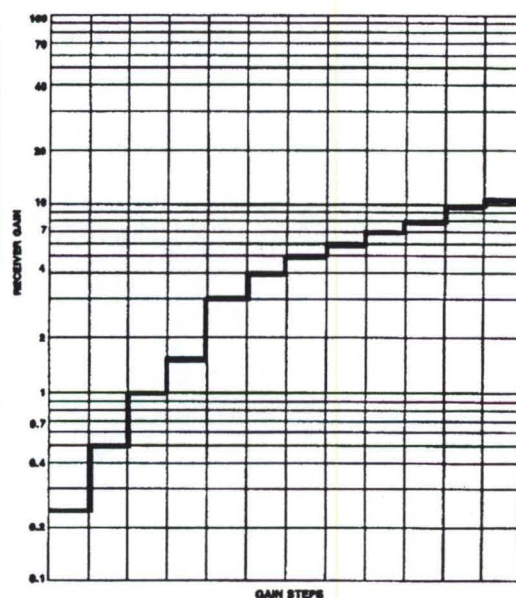


FIGURE 3: RECEIVER GAIN VS GAINSTEP NUMBERS

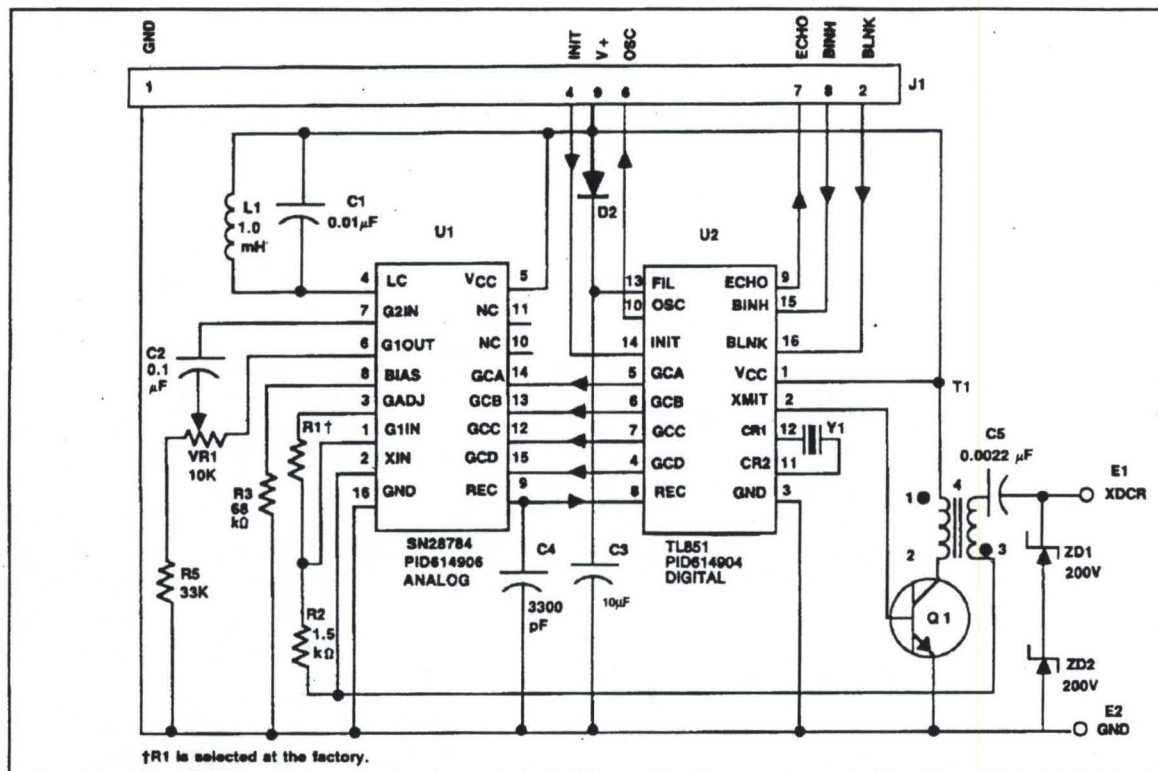


FIGURE 4: SCHEMATIC

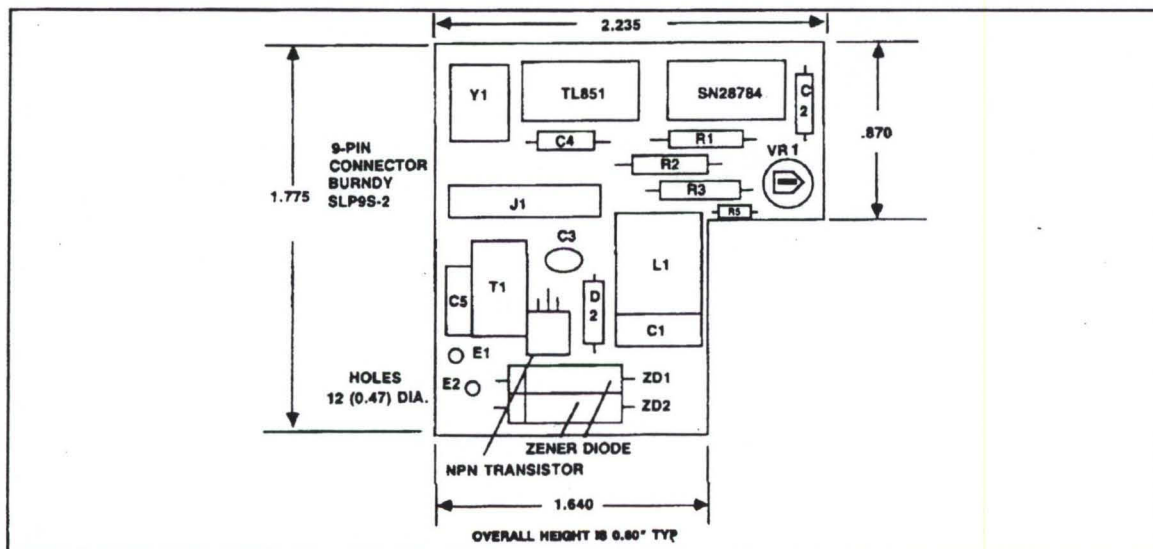


FIGURE 5: COMPONENT LAYOUT AND DIMENSIONS OF MODULE

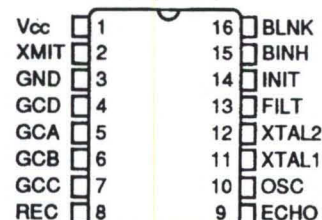
Technical Specifications for Digital Chip

Part# 616344-Minimum order 25, increments of 25Part# 614904-Minimum order 100. increments of 100

Features:

- Designed for Use with the Polaroid Analog Chip #614906 in Sonar Ranging Modules like the 6500 Series
- Operates with Single Supply
- Accurate Clock Output for External Use
- Synchronous 4-Bit Gain Control Output
- Internal 1.2-V Level Detector for Receive
- TTL-Compatible
- Interfaces to Electrostatic or Piezoelectric Transducers

N DUAL IN-LINE PACKAGE
(Top View)



Description:

The TL851 is an economical digital I2L ranging control integrated circuit designed for use with the Polaroid 614906 sonar ranging receiver integrated circuit. The TL851 is designed for distance measurement from 6" to 35 ft. The device has an internal oscillator that uses a low-cost external ceramic resonator. With a simple interface and a 420 kHz ceramic resonator, the device will drive a 50 kHz electrostatic transducer.

The device cycle begins when initiate (INIT) is taken to the high logic level. There must be at least 5 MS from initial power up (Vcc) to the first initiate signal in order for all the device internal latches to reset and for the ceramic resonator controlled oscillator to stabilize. The device will transmit a burst of 16 pulses each time INIT is taken high.

The oscillator output (OSC) is enabled by INIT. The oscillator frequency is the ceramic resonator frequency divided by 8.5 for the first 16 cycles (during transmit) and then the oscillator frequency changes to the ceramic resonator frequency divided by 4.5 for the remainder of the device cycle.

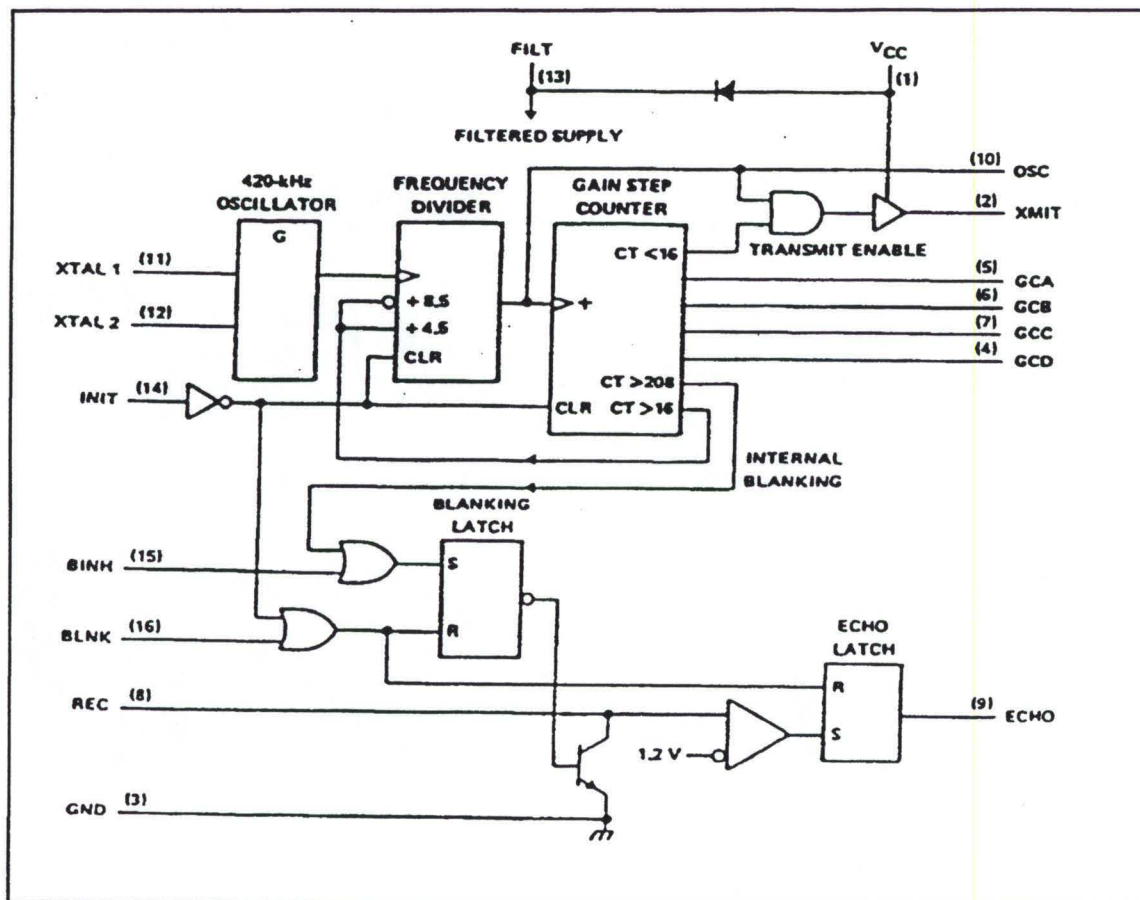
When used with an external 420 kHz ceramic resonator, the device internal blanking disables the receive input (REC) for 2.38 ms after initiate to exclude false receive inputs that may be caused by transducer ringing. The internal blanking feature also eliminates echoes from objects closer than 1.3 ft. from the transducer. If it is necessary to detect objects closer than 1.3 ft., then the internal blanking may be shortened by taking the blanking inhibit (BINH) high, enabling the receive input. The blanking input (BLNK) may be used to disable the receive input and reset ECHO to a low logic level at any time during the device cycle for selective echo exclusion or for a multiple-echo mode of operation.

The device provides a synchronous 4-bit gain control output (12 steps) designed to control the gain of the TL852 sonar ranging receiver integrated circuit. The digital gain control waveforms are shown in Figure 2 with the nominal transition times from INIT listed in the Gain Control Output Table.

The threshold of the internal receive level detector is 1.2 volts. The TL851 operates over a supply voltage range of 4.5 volts to 6.8 volts and is characterized for operation from 0°C to 40°C.

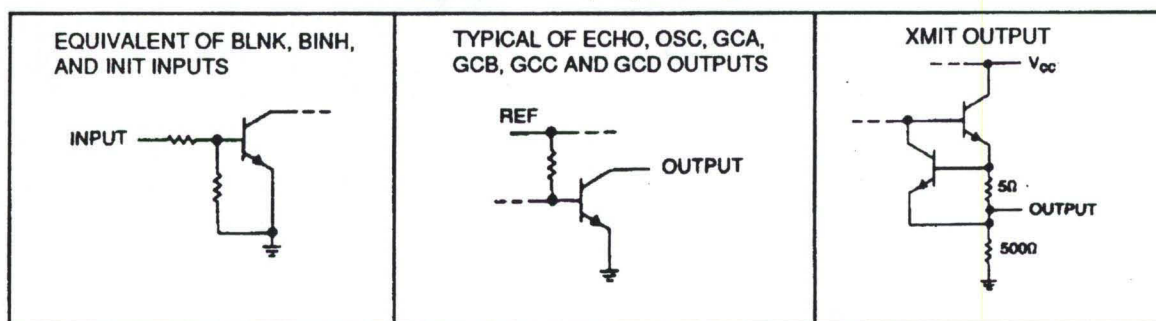
GAIN CONTROL OUTPUT TABLE					
STEP NUMBER	GCD	GCC	GCB	GCA	TIME (ms) FROM INITIATE*
0	L	L	L	L	2.38 ms
1	L	L	L	H	5.12 ms
2	L	L	H	L	7.87 ms
3	L	L	H	H	10.61 ms
4	L	H	L	L	13.35 ms
5	L	H	L	H	16.09 ms
6	L	H	H	L	18.84 ms
7	L	H	H	H	21.58 ms
8	H	L	L	L	27.07 ms
9	H	L	L	H	32.55 ms
10	H	L	H	L	38.04 ms
11	H	L	H	H	INIT ↓

*This is the time to the end of the indicated step and assumes a nominal 420 kHz ceramic resonator.



FUNCTIONAL BLOCK DIAGRAM

Schematic of Chip Inputs/Outputs



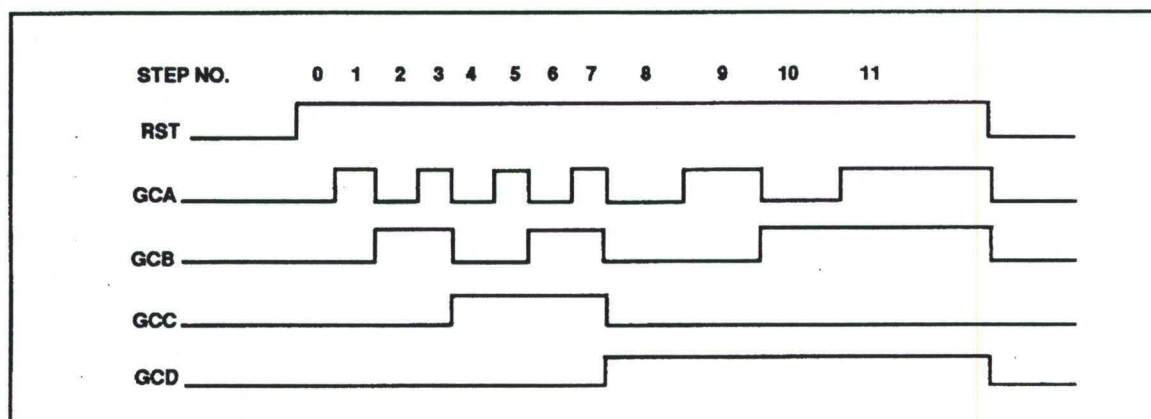


FIGURE 1: DIGITAL GAIN CONTROL WAVEFORMS

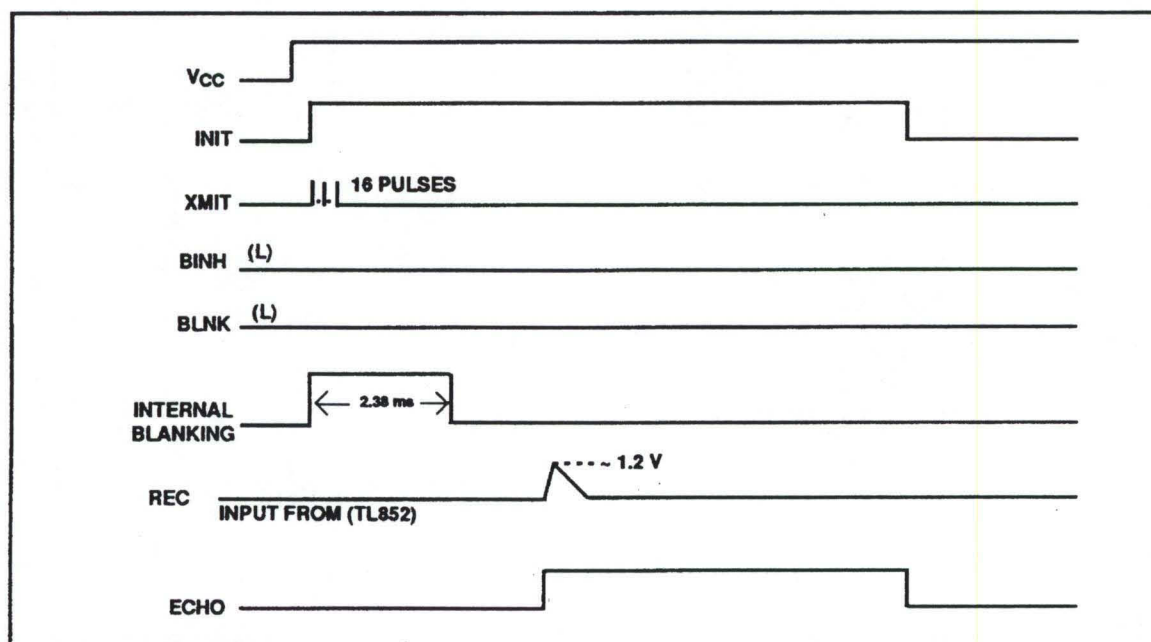


FIGURE 2: EXAMPLE OF A SINGLE-ECHO-MODE CYCLE WHEN USED WITH THE 614906 RECEIVER AND 420 kHz CERAMIC RESONATOR

Absolute Maximum Ratings Over Operating Free-air Temperature Range (unless otherwise noted):

Voltage at any pin with respect to GND.....	- 0.5 V to 7 V
Voltage at any pin with respect to Vcc.....	-7 V to 0.5 V
Continuous power dissipation at (or below) 25°C free-air temperature (see Note).....	1150 mW
Operating free-air temperature range	0° to 70°C
Storage temperature range	-65° to 150°C
Lead temperature 1.6 mm (1 /16 inch) from case for 10 seconds	260°C

Note 1: For operation above 25°C, derate linearly at the rate of 9.2mW/C

Recommended Operating Conditions

		MIN.	MAX.	UNIT
Supply Voltage, V_{CC}		4.5	6.8	V
High-level Input Voltage, V_{IH}	BLNK, BINH, INIT	2.1		V
Low-level Input Voltage, V_{IL}	BLNK, BINH, INIT		0.6	V
Delay Time, Power Up to INIT High		5		ms
Operating Free-air Temperature, T_A		0	40	°C

Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input Current	BLNK, BINH, INIT	$V_I = 2.1$ V			1	mA
High-level Output Current, I_{OH}	ECHO, OSC, GCA, GCB, GCC, GCD	$V_{OH} = 5.5$ V			100	μ A
Low-level Output Voltage, V_{OL}	ECHO, OSC, GCA, GCB, GCC, GCD	$I_{OL} = 1.6$ mA			0.4	V
On-state Output Current	IMIT output	$V_O = 1$ V		-140		mA
Internal Blanking Interval	REC input			2.38†		ms
Frequency During 16-pulse	OSC output			49.4†		kHz
Transmit Period	XMIT output			49.4†		
Frequency After 16-pulse	OSC output			93.3†		kHz
Transmit Period	XMIT output			0		
Supply Current, I_{CC}	During transmit period				260	mA
	After transmit period				55	

†These typical values apply for a 420 kHz ceramic resonator.

Technical Specifications for

Analog Chip Ref: SN28784N

Part# 616346-Minimum order 25, increments of 25....Part# 614906-Minimum order 100, increments of 100

Absolute Maximum Ratings:

V_{CC} (PIN 5 to 16): 7.0 V
 Voltage from any PIN to GRD:.....-0.5,+7 V
 Voltage from any PIN to V_{CC}:..... -0.5V Operating
 Temperature Range: -40 to 85° Storage
 Temperature Range: 65 to 150° PIN
 2 Input Current (50% Duty Cycle): ±60 mA

Features:

Economical sonar receiver 10 for use with Polaroid's Digital IC. Digitally controlled exponential gain/Q control maximizes noise and side lobe rejection. Overall gain adjustable with one external resistor. Designed to work in the 50 kHz range, but may be adapted to other frequencies from 20 to 90 kHz. Requires a minimum of external components.

PINOUT Description:

PIN	Name	Description
1	G1 IN	Receiver input
2	XIN	TRANSDUCER input
3	GADJ	GAIN ADJUSTMENT
4	LC	Tuned circuit
5	V _{CC}	Positive supply
6	GIOUT	FIRST GAIN stage OUTPUT
7	G2IN	SECOND GHAIN stage INPUT
8	BIAS	BIAS resistor
9	REC	Detected echo output
10	NC	NO internal CONNECTION
11	NC	NO internal CONNECTION
12	GCC	GAIN CONTROL 0
13	GCB	GAIN CONTROL B
14	GCA	GAIN CONTROL A (LSB)
15	GCD	GAIN CONTROL D (MSB)
16	GNC	Circuit GROUND

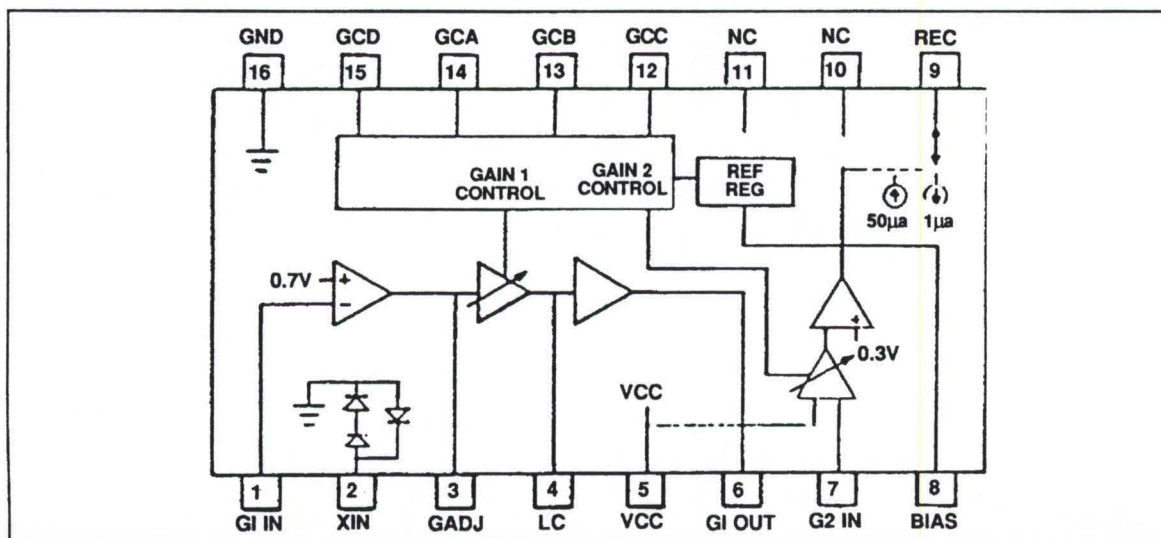


FIGURE 1: BLOCK DIAGRAM

ELECTRICAL SPECIFICATION:

PARAMETER:	MIN.	TYP.	MAX.	UNITS
Recommended Operating Conditions:				
Supply voltage (Vcc)	4.5	5.0	6.8	V
GCA, GCB, GCC, and GCD inputs:				
Vil			0.6	V
Vih	2.0			V
Operating temperature	0		40	°C
Bias resistor (between pin 8 and 16)	64.6		71.4	Kohms
Electrical Characteristics (0 to 40°C):				
Continuous operating current			45	mA
GCA, GCB, GCC, and GCD inputs:				
Iil at Vin = 0V, Vcc = 5V			-3	mA
Iih at Vin = 2V, Vcc = 5V		-0.5		mA
V open circuit Vcc = 5V		2.5		V
Receive output sink current at 1 V (no signal input)		1		μA
Receive output source current at 1 V (during detect)		50		μA
Current through LC network			2.0	mA
XIN positive clamp at Iin=40 mA			2.5	V
XIN negative clamp at Iin=40 mA			1.3	V

DETECT LEVEL vs GAIN STEP:

See tests circuit in FIGURE 4. Detect level is measured by applying a 15 cycle burst of 49.4 kHz square wave just after the beginning of the gain step to be tested. The least burst amplitude which makes REC pin reach 1.2V is recorded as the "detect level". System "gain" is then inversely proportional to "detect level".

Step Number	GCA	GCB	GCC	GCD	Typical Relative Gain* (Peak to Peak)
0	0	0	0	0	See Figure 5
1	1	0	0	0	
2	0	1	0	0	
3	1	1	0	0	
4	0	0	1	0	
5	1	0	1	0	
6	0	1	1	0	
7	1	1	1	0	
8	0	0	0	1	
9	1	0	0	1	
10	0	1	0	1	
11	1	1	0	1	

**Assumes nominal external components and typical IC*

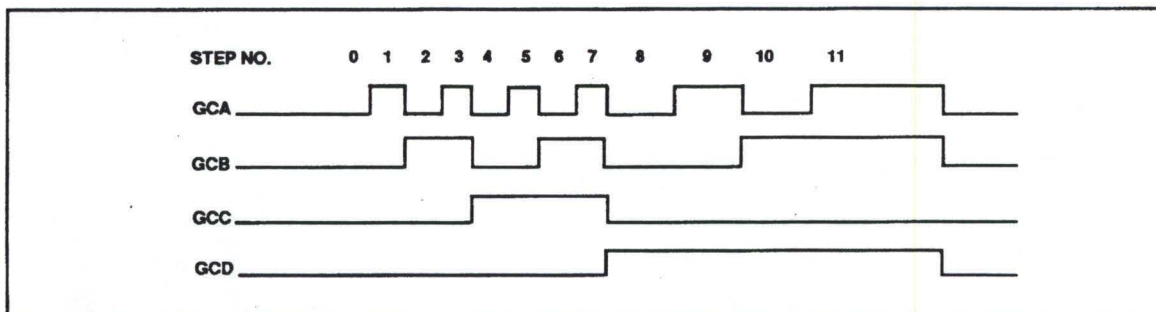


FIGURE 2: TYPICAL DIGITAL GAIN CONTROL WAVEFORMS *Note: The counters providing the gain signals must be synchronous.*

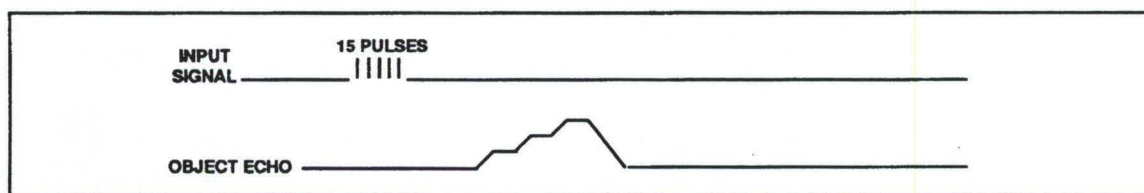


FIGURE 3: EXAMPLE OF THE REC PIN RESPONSE TO A 15 CYCLE BURST (TEST CIRCUIT IN FIGURE 4)

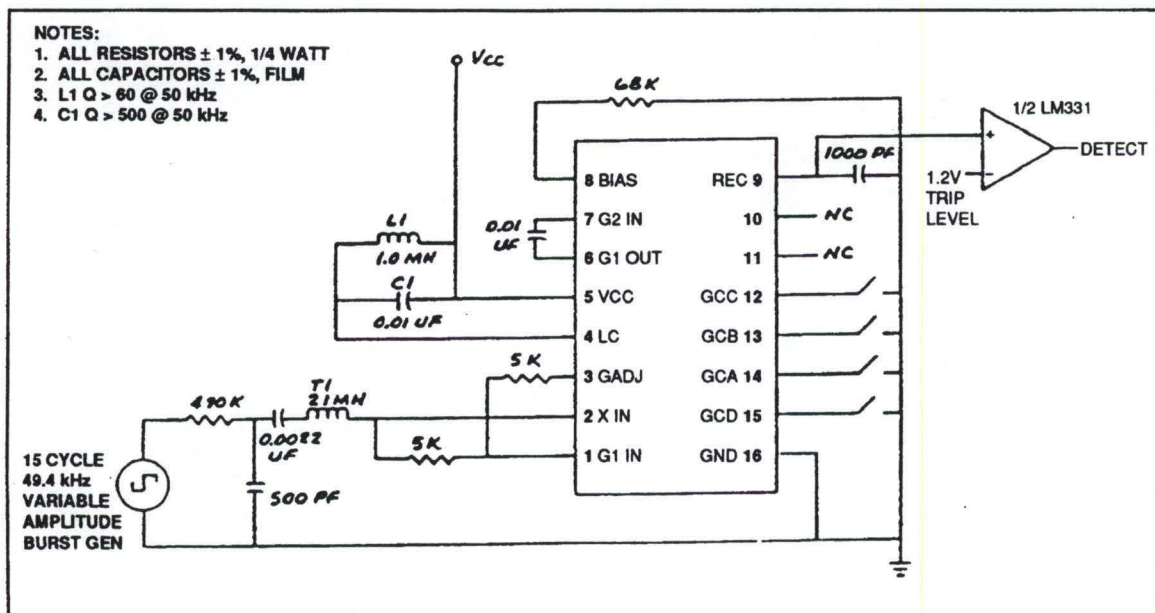


FIGURE 4: DIRECT LEVEL MEASUREMENT CIRCUIT

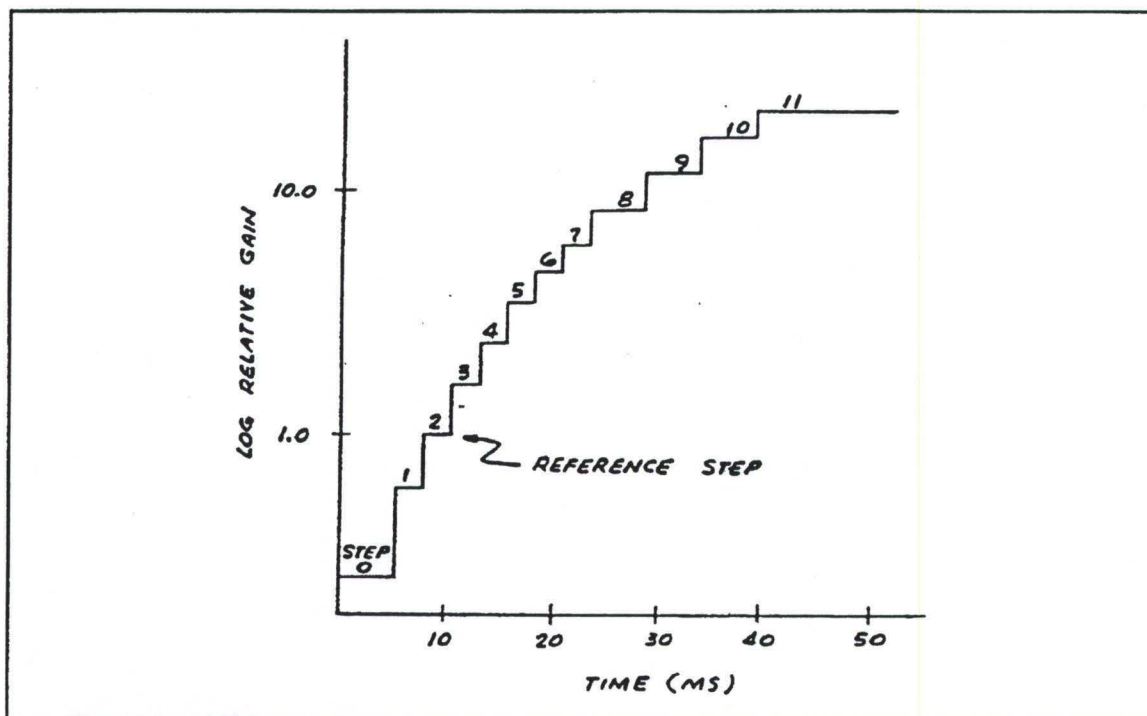


FIGURE 5: TYPICAL LOG RELATIVE GAIN VS GAIN CONTROL STEPS

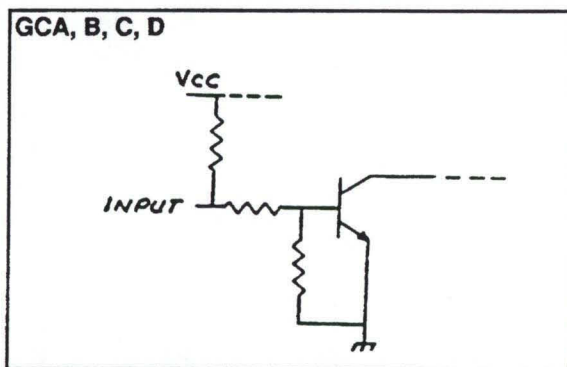


FIGURE 6: TYPICAL INPUT SCHEMATIC